TECHNISCHE UNIVERSITÄT
ILMENAU

Integrated Communication Systems Group

M.Sc. Oleksandr Andryeyev

Aerial Base Station Placement in Temporary-Event Scenarios

Dissertation to obtain the academic degree Doktoringenieur (Dr.-Ing.)

30 April 2020

Please cite as:
M.Sc. Oleksandr Andryeyev, "Aerial Base Station Placement in Temporary-Event Scenarios,"
Dissertation (Doktorarbeit), Technische Universität Ilmenau, Department of Computer Science and Automation,
April 2020.

urn:nbn:de:gbv:ilm1-2020000464

Technische Universität Ilmenau
Department of Computer Science and Automation
Integrated Communication Systems Group

Helmholtzplatz 5 · 98693 Ilmenau · Germany
http://www.tu-ilmenau.de/iks

Aerial Base Station Placement in Temporary-Event Scenarios

DISSERTATION

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der
Fakultät für Informatik und Automatisierung
der Technische Universität Ilmenau

von
M.Sc. Oleksandr Andryeyev

geboren am 10. Juni 1990
in Kharkiv, Ukraine

Gutachter:

1. Univ.-Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel, Technische Universität Ilmenau
2. Univ.-Prof. Dr.-Ing. habil. Armin Zimmermann, Technische Universität Ilmenau
3. Prof. Dr.-Ing. Jens Mückenheim, Hochschule Merseburg

Tag der Einreichung: 30. April 2020

Tag der wissenschaftlichen Aussprache: 20. Oktober 2020

urn:nbn:de:gbv:ilm1-2020000464

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt.

Ich bin darauf hingewiesen worden, dass die Unrichtigkeit der vorstehenden Erklärung als Täuschungsversuch bewertet wird und gemäß § 7 Abs. 10 der Promotionsordnung den Abbruch des Promotionsverfahrens zur Folge hat.

(M.Sc. Oleksandr Andryeyev)
Ilmenau, 30 April 2020

Acknowledgements

I would like to thank my supervisor, Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel for his patience, encouragement and scientific liberty. During my doctoral studies he gave me a lot of support and constructive feedback in my research. Besides my supervisor, I would like to thank my co-supervisor Dr.-Ing. Oleksandr Artemenko for his support and helpful discussions, especially in the initial phase. Without their guidance, constructive and collegial cooperation, this research could not have been successfully conducted.

I am eternally grateful to my loving parents Valentyna and Yuriy, my sincere brother Igor for their support throughout the years of study. I'd like to express my deepest gratitude to my beloved wife Mariia for her unstinting support, unconditional love and patience.

My sincere thanks also goes to Evgeniy, Oleg und Vasiliy for being wonderful company during all these years and plentiful discussions of scientific and non-scientific matter. I would like also to thank all my friends whom I met in Ilmenau, Dmitry, Sergiy and Oleksiy, Yauheniya, Abebe, Lilli and others.

I want to express my gratefulness to the stuff members of the ICS group for their constant support during my research work, exclusively Dr.-Ing. Karsten Henke, Dr.-Ing. Heinz- Dietrich Wuttke, Nicole Sauer, Jürgen Schmidt, Victor Casas, Tobias Simon, Andre Puschmann, Mohamed Kalil, Alina Rubina, Abubaker Waswa and others. I am especially thankful for the critical feedback received during our discussions. It was an honor to me to work with you within our research group.

Finally, I am highly grateful for the reviewers of this thesis and for the promotion committee.

Abstract

As the traffic demands have grown dramatically in recent years, so has the interest in developing novel solutions that increase the network capacity in cellular networks. The problem of capacity improvement is even more complex when applied to a dynamic environment during a disaster or temporary event. The use of aerial base stations has received much attention in the last ten years as the solution to cope with the dynamics of the changing environment and to supplement the ground infrastructure with extra capacity. Due to higher elevations and possibility to place aerial base stations in close proximity to the user, path loss is significantly smaller in comparison to the ground infrastructure, which in turn enables high data capacity.

We are studying the optimization problem of maximizing network capacity by proper placement of aerial base stations. To handle the changes in the dynamic radio environment, it is necessary to promptly solve the optimization problem. However, we show that the optimal placement of aerial base stations is the NP-hard problem and its solution is non-trivial, and thus, there is a need for fast and scalable optimization algorithms.

This dissertation investigates how to solve the placement problem efficiently and to support the dynamics of temporary events. First, we propose a novel hybrid algorithm (Projected Clustering), which calculates multiple solutions based on the fast distance-based capacity approximation and evaluates them on the accurate SINR-based capacity model, avoiding sub-optimal solutions. Second, we propose a novel distributed, self-organized framework (AIDA), which conducts a decision-making process using only local knowledge, decreasing the network overhead and relaxing the requirements for communication between aerial base stations.

During the formulation of the placement problem, we found that there is still considerable uncertainty with regard to air-to-ground propagation modeling. Since this aspect plays an important role in our analysis, we validated state-of-the-art air-to-ground propagation models by collecting real measurements and chose the most accurate model for the simulations.

Simulation results show that user distribution plays a significant role in the placement optimization problem and state-of-the-art algorithms generate sub-optimal solutions in the case of multiple dense user clusters. The proposed centralized Projected Clustering algorithm takes into account the problem-specific knowledge in a form of user distribution analysis. As a result, it outperforms state-of-the-art algorithms, being $\sim 89x$ faster and yielding $\sim 31\%$ more capacity.

Finally, we show that the proposed self-organized AIDA framework can further improve the scalability of the centralized algorithms and significantly reduce the computational time (around $100x$), while achieving nearly the same solution quality as the centralized algorithms (around 10% less capacity). We have found that tight coupling between non-cooperative agents in the self-organized framework leads to reduction of agent performance.

Kurzfassung

Die Anforderungen an den Netzdatenverkehr sind in den letzten Jahren dramatisch gestiegen, was ein großes Interesse an der Entwicklung neuartiger Lösungen zur Erhöhung der Netzkapazität in Mobilfunknetzen erzeugt hat. Besonderes Augenmerk wurde auf das Problem der Kapazitätsverbesserung bei temporären Veranstaltungen gelegt, bei denen das Umfeld im Wesentlichen dynamisch ist. Um der Dynamik der sich verändernden Umgebung gerecht zu werden und die Bodeninfrastruktur durch zusätzliche Kapazität zu unterstützen, wurde der Einsatz von Luftbasisstationen vorgeschlagen. Die Luftbasisstationen können in der Nähe des Nutzers platziert werden und aufgrund der im Vergleich zur Bodeninfrastruktur höheren Lage die Vorteile der Sichtlinienkommunikation nutzen. Dies reduziert den Pfadverlust und ermöglicht eine höhere Kanalkapazität.

Das Optimierungsproblem der Maximierung der Netzkapazität durch die richtige Platzierung von Luftbasisstationen bildet einen Schwerpunkt der Arbeit. Es ist notwendig, das Optimierungsproblem rechtzeitig zu lösen, um auf Veränderungen in der dynamischen Funkumgebung zu reagieren. Die optimale Platzierung von Luftbasisstationen stellt jedoch ein NP-schweres Problem dar, wodurch die Lösung nicht trivial ist. Daher besteht ein Bedarf an schnellen und skalierbaren Optimierungsalgorithmen.

Als Erstes wird ein neuartiger Hybrid-Algorithmus (Projected Clustering) vorgeschlagen, der mehrere Lösungen auf der Grundlage der schnellen entfernungsbasierten Kapazitätsapproximierung berechnet und sie auf dem genauen SINR-basierten Kapazitätsmodell bewertet. Dabei werden suboptimale Lösungen vermieden. Als Zweites wird ein neuartiges verteiltes, selbstorganisiertes Framework (AIDA) vorgeschlagen, welches nur lokales Wissen verwendet, den Netzwerkmehraufwand verringert und die Anforderungen an die Kommunikation zwischen Luftbasisstationen lockert.

Bei der Formulierung des Platzierungsproblems konnte festgestellt werden, dass Unsicherheiten in Bezug auf die Modellierung der Luft-Bodensignalausbreitung bestehen. Da dieser Aspekt im Rahmen der Analyse eine wichtige Rolle spielt, erfolgte eine Validierung moderner Luft-Bodensignalausbreitungsmodelle, indem reale Messungen gesammelt und das genaueste Modell für die Simulationen ausgewählt wurden.

Simulationsergebnisse zeigen, dass die Benutzerverteilung eine wichtige Rolle bei der Platzierungsoptimierung spielt und modernste Algorithmen bei mehreren dichten Benutzerclustern suboptimale Lösungen generieren. Der vorgeschlagene zentralisierte Projected-Clustering-Algorithmus berücksichtigt das problemspezifische Wissen in Form einer Analyse der Benutzerverteilung. Dadurch übertrifft es modernste Algorithmen, ist 89 Mal schneller und bietet 31% mehr Kapazität.

Das vorgeschlagene selbstorganisierte AIDA Framework kann zudem die Skalierbarkeit der zentralisierten Algorithmen verbessern und die Rechenzeit ($100x$) signifikant reduzieren, während es nahezu die gleiche Lösungsqualität wie die zentralisierten Algorithmen erreicht (ca. 10% weniger Kapazität). Weiterhin konnte festgestellt werden, dass eine enge Kopplung zwischen nicht-kooperativen Agenten im selbstorganisierten Framework zu einer Verringerung der Leistung des Agenten führt.

List of Publications

I Oleksandr Andryeyev, Umut Onus, Victor Casas Melo, Andreas Mitschele-Thiel: Experimental Validation of Air-to-Ground Propagation Models for Low-Altitude Platforms, 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), WiDroIt 2019, Santorini Island, Greece, May 2019.

II Oleksandr Andryeyev, Andreas Mitschele-Thiel: Efficiency vs. Accuracy of Aerial Base Station Placement, International Conference on Networked Systems 2019 (NetSys 2019), Garching b. München, Germany, March 2019.

III Oleksandr Andryeyev, Andreas Mitschele-Thiel: Increasing the Cellular Network Capacity Using Self-Organized Aerial Base Stations, 3rd Workshop on Micro Aerial Vehicle Networks,Systems, and Applications (DRONET 17), MobiSys 2017, ACM, Niagara Falls, New York, USA, June 2017.

IV Alina Rubina, Oleksandr Artemenko, Oleksandr Andryeyev, Andreas Mitschele-Thiel: A Novel Hybrid Path Planning Algorithm for Localization in Wireless Networks , 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DRONET 17), MobiSys 2017, ACM, pp.13-16 , Niagara Falls, New York, USA, June 2017.

V Alina Rubina, Oleksandr Andryeyev, Mehdi Harounabadi, Ammar Al-Khani, Oleksandr Artemenko, Andreas Mitschele-Thiel: Investigation and Adaptation of Signal Propagation Models for a Mixed Outdoor-Indoor Scenario Using a Flying GSM Base Station, 8th EAI International Conference on Ad Hoc Networks, Ottawa, Canada, September 2016.

VI Oleksandr Artemenko, Omachonu Joshua Dominic, Oleksandr Andryeyev, Andreas Mitschele-Thiel: Energy-Aware Trajectory Planning for the Localization of Mobile Devices Using an Unmanned Aerial Vehicle , The 25th International Conference on Computer Communication and Networks (ICCCN 2016), pp. 1-9. , Waikoloa, Hawaii, USA, August 2016.

VII Thomas Dietrich, Oleksandr Andryeyev, Armin Zimmermann, Andreas Mitschele-Thiel: Towards a Unified Decentralized Swarm Management and Maintenance Coordination Based on MAVLink , IEEE Int. Conf. on Autonomous Robot Systems and Competitions (ICARSC 2016), pp. 124-129 , May 2016.

VIII Oleksandr Andryeyev, Alina Rubina, Oleg Golokolenko, Oleksandr Artemenko, Andreas Mitschele-Thiel: SkySAIL: a Flexible Software-Defined Radio Enabled Micro Aerial Vehicle, Tenth IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2016), Waikoloa, Hawaii, USA, August 2016.

IX Oleksandr Andryeyev, Oleksandr Artemenko, Andreas Mitschele-Thiel: Improving the System Capacity Using Directional Antennas With a Fixed Beam on Small Unmanned Aerial Vehicles, EUCNC 2015, Paris, France, June 2015.

X Oleksandr Andryeyev: Spectrum Usage Improvement in Cognitive Radio Networks Using Directional Antenna and UAV Capabilities, SPITSE 2014, Ilmenau, Germany, July 2014.

Contents

Chapter 1

Introduction

In recent years, the emergence of a variety of mobile devices and their applications has caused a rapid growth of the volume of mobile traffic carried by cellular networks. For this reason, network operators are forced to search for solutions to significantly increase network capacity with limited spectrum. The problem of capacity improvement is more complex when applied to a dynamic environment during a disaster [1] or a temporary event [2]. Temporary events can be characterized by unexpected and drastically increased load, congested network and widely varied network conditions. Besides, over the event duration, users may change their positions, changing the spatial location of the traffic demand. The resulting dynamics of network traffic provide the motivation to develop traffic offloading solutions that can dynamically adapt to the variation in cost-effective manner.

Advances in technologies brought us cheap, small and powerful unmanned aerial platforms which provide base station functionality [2,3]. The use of Aerial Base Stations (ABSs) has received much attention in the last ten years as an option to temporarily serve areas with high data demand. ABSs can be employed to identify traffic demand and support its offloading by using the key functionality of unmanned aerial platforms – the capability of controlling their positions. Using this capability, ABSs can be deployed in close proximity to the users and dynamically adapt to the variation in traffic distribution over space. Hence fast and scalable optimization algorithms are needed, which can promptly solve the placement problem in real-time or near real-time. However, the optimal placement of ABSs is a NP-hard problem and its solution is non-trivial.

1.1 Problem Statement and Scope of Research

The problem to be solved is as follows:

- the metrics for performance evaluation are: the aggregated communication system capacity over ABSs, which has to be maximized; the execution time of the algorithm, which has to be minimized;
- a set of aerial base stations with a fixed number of available spectrum resources (resource blocks) is given;
- a set of mobile users (aka User Equipments (UEs)) of any distribution is given a priori;
- at run-time, the ABSs adapt their positions to offload the traffic of UEs;

The central research questions are as follows:

1. How can the aerial base station placement problem be modeled and formulated as an optimization problem?
2. To what extent can the use of problem-specific knowledge of the aerial base station placement problem improve the performance of centralized optimization algorithms?
3. Is it possible to design a scalable self-organized aerial base station placement framework, which uses only locally available information? What are the quantitative and qualitative characteristics of such an approach and how close can it come to centralized placement algorithms in terms of the system performance?
4. To what extent do the existing air-to-ground radio propagation models correspond to the real air-to-ground communication channel?

1.2 Contributions of the Thesis

Aerial communication system performance depends on multiple design decisions, such as frequency resource allocation and trajectory planning strategies. We show that the network capacity can be improved by spatial resource reuse and the controlled movement of ABSs. This work is presented in Chapter 2 and has been published previously in [4,5].

To model radio propagation, multiple air-to-ground propagation models for low-altitude communication systems have recently been proposed in the literature. These models have been developed using analytical derivations or ray-tracing simulations and were not experimentally validated. To validate and choose the most suitable propagation model, a low-altitude aerial base station has been designed and implemented on a small micro-aerial vehicle. Using this prototype, we obtained realistic measurements

of the air-to-ground communication channel. We found that only one of the available state-of-the-art models suits our measurements of the radio propagation channel with high degree of accuracy, while two other models do not show a good correlation with the experimental data. This is described in Chapter 5 and has been published previously in [6].

We are studying centralized algorithms to optimize the placement of aerial base stations to maximize the system capacity. We compare solutions with respect to quality as well as computational complexity, i.e. the time required to derive the placement. We show that our algorithm, employing a distance-based capacity approximation function, outperforms previous proposals by a factor of up to 100 in terms of computational time without any loss in quality of the solution. These results are presented in Chapter 3 and have been published previously in [7].

Aerial base station deployment is a dynamic process. Since users are mobile, the placement problem changes continuously and the optimal solution will differ in time. Therefore, the optimization process needs to be repeated continuously. This inevitably results in high computing and communication requirements in the case of a centralized approach. We present a self-organized framework to address this. We compare the developed framework with centralized approaches in terms of available system capacity, adaptability and fault-tolerance. The self-organized approach shows a good performance in comparison to the centralized approaches with only small capacity degradation. We show that this approach works well in the case of no or weak coupling between self-organized agents and propose a method to ensure their decoupling. This work is presented in Chapter 4 and has been partially published previously in [8].

1.3 Thesis Organization and Outline

The thesis is organized as follows.

Chapter 2 introduces the aerial base station placement problem, presents the related work and assumptions for the system model.

As a first step, Chapter 3 gives a formulation of the placement problem as an optimization problem and discusses its properties. Here the state-of-the-art approaches and algorithms for the ABS placement problem are reviewed in detail. It concludes with a proposal of a novel centralized algorithm for the placement problem.

Chapter 4 discusses the discovered issues of centralized optimization algorithms for the ABS placement problem and proposes to use a novel self-organized approach to improve scalability and fault-tolerance. Then we discuss its quantitative and qualitative properties in comparison to centralized solutions.

Chapter 5 introduces the experimental setup and the experimental validation of the radio propagation model. We compare our measurements to several air-to-ground radio propagation models from the literature.

Chapter 6 concludes this work and presents possible future work.

Additional information about the developed ABS prototype and the simulation environment can be found in Appendix.

Chapter 2

Background and Related Work

> All models are wrong, but some are useful.
>
> George Box, British statistician

In this chapter, we start with a brief overview on related work in ABS placement and describe the main goals, problems and solutions that have been proposed. We then describe the system model that is used in this work.

2.1 Related Work in Aerial Base Station Placement

Multiple studies show that ABSs can be used to complement existing cellular infrastructures in disasters or during temporary events [9–11]. They show that ABSs can provide better Line-of-Sight (LoS) conditions due to higher elevations in comparison to conventional Base Stations (BSs) installed on roofs of buildings and can usually be deployed at almost any point in the area.

However, the placement of ABSs is a complex and challenging task. Inappropriate placement can cause the channel to degrade due to increased path loss or interference levels. Thus, significant research efforts have been made to understand and address the ABS placement problem. Table 2.1 briefly summarizes the related work and compares it with the approaches proposed in this thesis. The related work in the emerging area of ABS placement has a certain level of uncertainty and ambiguity. The table attempts to sort the related work by scenario constraints and optimization objectives. In addition, we highlight which methods are used in the related work and which assumptions are made during modeling.

Table 2.1. – Related work in aerial base station placement.

Reference	Problem					Methods			Approximations			
	Scenario Constraint	Objective	# ABSs	Ground BSs	Frequency Reuse 1	Algorithm	Class of Algorithm	Comp. Complexity	Prop. Model	Downlink/Uplink	Interference	Model Approx.
Al-Hourani et al. [9]	AC	PL	1	✗	✗	n/a	C	-	A2G-1	DL	✗	✗
Mozaffari et al. [12]	AC	CAP	1	✗	-	n/a	C	-	A2G-1	DL	I	✗
Mozaffari et al. [13]	AC	PL	n	✗	✓	n/a	C	-	A2G-2	DL	A	✗
Alzenad et al. [14]	UC	P	1	✗	-	Gradient descent	C	M	A2G-1	DL	✗	✗
Shakhatreh et al. [15]	UC	P	1	✗	-	PSO	C	H	ITU,OI	DL	✗	✗
Bor-Yaliniz et al. [16]	UC	UN	1	✗	-	Interior-point optimization	C	H	A2G-1	DL	✗	✗
Lyu et al. [17]	UC	N	n	✗	✗	Sequential spiral placement	C	M	Friis	DL	✗	✗
Mozaffari et al. [18]	UC	P	n	✗	✗	K-Means	C	L	A2G-1	UL	✗	✓
Kalantari et al. [19]	UC	UN	n	✗	✓	PSO	C	H	A2G-1	DL	A	✗
Mozaffari et al. [20]	UC	L	n	✗	✓	Truncated octahedron cells, bisection search	C	H	A2G-1	DL	A	✓
Sharma et al. [21]	TO	CAP	1	✓	-	Cost-based neural network	C	H	Friis	DL	✗	✗
Ghanavi et al. [22]	TO	CAP	1	✓	-	Q-Learning	C	H	A2G-1	DL	✗	✗
Bor-Yaliniz et al. [23]	TO	UN	1	✗	-	Interior-point optimization	C	H	A2G-1	DL	✗	✗
Sharma et al. [24]	TO	L	n	✓	✓	Entropy neural network	C	H	Friis	UL, DL	A	✗
Zhong et al. [25]	TO	CAP	n	✓	✗	Virtual Forces + K-Means	C	L	A2G-1	DL	✗	✓
Galkin et al. [26]	TO	CAP	n	✓	✗	K-Means	C	L	3GPP, LoS	DL	✗	✓
Rohde et al. [27]	TO	CAP	n	✓	✓	Study (predefined positions)	C	-	Friis	DL	A,I	✗
Kalantari et al. [28]	TO	CAP	n	✓	✗	PSO	C	H	A2G-1	DL	✗	✗
Our Approach, Ch. 3	TO	CAP	n	✗	✓	PC	C	M	A2G-1	DL	A	✓
Our Approach, Ch. 4	TO	CAP	n	✗	✓	SO AIDA	SO	L	A2G-1	DL	A	✓

AC = Area Coverage
UC = User Coverage
TO = Traff. Offloading

PL = Path Loss
CAP = Capacity
UN = User Number
P = Power
N = Number of ABSs
L = Latency

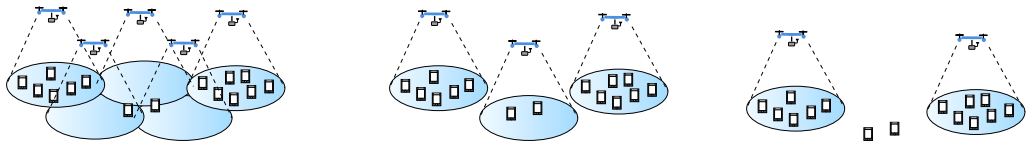
n/a = no details in the paper

C = Centralized
SO = Self-Organized
H = High, L = Low
M = Moderate

A2G-1: [9]
A2G-2: [29]
OI = Outdoor-Indoor

DL = Downlink
UL = Uplink

A = ABS-ABS
I = ABS-Infrastructure



(a) Complete area coverage (AC) (b) Complete user coverage (UC) (c) No constraints, traffic offloading (TO)

Figure 2.1. – ABS placement optimization: examples of placements under specific scenario constraints.

Historically, ABSs have been used in military missions for battlefield network deployment [30] and for network deployment in disaster scenarios, where existing cellular infrastructure is destroyed [31]. In these applications, UEs require communication in all geographical areas. While the location of UEs and their number are often unknown in advance, the main problem is to provide wireless coverage for the entire disaster area as shown in Fig 2.1(a). In [9] authors investigated this problem for a single ABS. They analyzed a tradeoff between the geographical coverage and the path loss between the ABS and the UEs. At lower altitudes, the signal attenuates more due to reflections and scattering; at higher altitudes, the signal attenuates more due to the increased path loss between the ABS and the UEs. Mozaffari et al. [12] extended this work and formulated the optimization problem for a single ABS to minimize the path loss between the ABS and the UEs, while ensuring the coverage for the entire disaster area. For multiple ABSs, which use the same frequency resources, the problem is more complex. In [13] the authors determined the locations of ABSs to maximize the total coverage area while ensuring that the coverage areas of ABSs do not overlap.

In case that user locations are known, the coverage can be provided exclusively for the regions containing users as shown in Fig 2.1(b). In this case we can reduce the number of required ABSs by focusing on the regions where UEs are present. However, multiple solutions of the placement problem exist that satisfy the constraint of guaranteeing user coverage. In [14, 15] authors minimized the transmission power of a single ABS under this constraint. Bor-Yaliniz et al. [16] proposed the method to find the placement, which maximizes the number of UEs with satisfied Quality of Services (QoSs) requirements. Lyu et al. [17] discussed the placement of multiple ABSs. The authors tried to minimize the number of ABSs required and proposed an algorithm, in which the ABSs are placed sequentially along a spiral path. The advantage of this algorithm is that the computational time can be estimated due to the deterministic spiral path. In [18] authors used K-Means clustering approach to find the placement, which minimizes the transmission power of multiple ABSs. They assumed that the transmission power

is proportional to the distance between the ABSs and the UEs. Kalantari et al. [19] suggested that significant capacity gain would be achieved if ABSs reused the radio spectrum in space. The authors proposed to use the Particle Swarm Optimization (PSO) to find the placement which maximizes the number of UEs with satisfied QoS requirements and minimizes the downlink interference created by the ABSs. In [20] authors discussed the concept of three-tier drone-cellular network that incorporated drone-UEs, low-altitude ABSs and high-altitude ABSs. High-altitude ABSs were assumed to provide a wireless backhaul connectivity. The goal was to find the placement, which ensures the coverage of drone-UEs and minimizes the average end-to-end latency at drone-UEs. According to the authors, the latency depends on the load of low-altitude ABSs and backhaul transmission rate.

In traffic offloading scenarios it is not always necessary to provide coverage for all UEs. Instead, the goal is to optimize the selected utility metric, and for this, an optimization algorithm also needs to find the best possible user subset as shown in Fig 2.1(c). Sharma et al. [21] tried to maximize the offloading capacity of a single ABS without the constraint of complete user coverage. They showed that even for the single ABS the problem is computationally very expensive and proposed to use a neural network to learn the solution empirically. However, the neural network needs a lot of time for learning and in the case of changes in the environment, it is necessary to retrain the neural network, which is associated with high data acquisition time and computing costs. Ghanavi et al. [22] proposed to use reinforcement learning (Q-learning) for the same problem. However, in a quickly-changing temporary event scenario, it is not realistic to try out many possible locations, because the learned information can be quickly outdated due to movement of the users. In [23] the authors tried to maximize the number of UEs served by a single ABS. Sharma et al. [24] suggested that the allocation of ABSs to the demand area introduces latency in the network. They proposed to minimize this latency using entropy neural networks, which learn the traffic demand over network operational time. Zhong et al. [25] investigated the problem of maximizing the offloading capacity of multiple ABSs. They proposed to set the initial ABSs locations using K-means and then apply the Virtual Fields (VF) concept to continuously adjust their positions. They assumed that the UEs generate the attraction forces for the ABS and that the ABSs use orthogonal frequency resources. Galkin et al. [26] suggested that in the case of multiple ABSs, the problem can be solved by the K-Means algorithm. However, they assumed that they know the number of user clusters in advance and that the ABSs use orthogonal frequency resources. Rohde et al. [27] studied the effects of interference in an irregularly loaded system induced by ABSs deployment in reuse 1 cellular networks. They showed that ABSs can significantly increase the traffic offloading capacity in this

scenario. Finally, Kalantari et al. [28] investigated the placement problem of multiple ABSs. They proposed to use PSO to solve this complex problem. However, the PSO algorithm is associated with a high computational cost.

In this thesis, the goal is to find the ABS placement, which maximizes the aggregated system capacity for the UEs served by the ABS system in the case of an overload of ground infrastructure. We consider multiple ABSs, which do not share radio resources with ground infrastructure. However, we assume that ABSs reuse resources among each other as it can significantly increase the offloading capacity [19, 20, 24, 27]. In contrast to work of others, we study how the ABS placement affects the downlink interference at UEs for various user distributions.

We analyze in details the computational complexity of the ABS placement and state-of-the-art approaches [18,26], which use clustering to solve the problem in a computationally efficient way. We show that in the conditions of resource reuse among ABS, this approach can generate suboptimal solutions because it does not take into account the impact of downlink interference. We propose a superior approach, which accounts for the downlink interference and is capable of generating near-optimal solutions in a computationally efficient way by employing a fast distance-based capacity approximation function. Moreover, we compare the proposed algorithm to clustering, direct-search and meta-heuristic optimization algorithms.

We also present a novel self-organized approach to further decrease computational time and to reduce the need in global knowledge with minor loss in solution quality.

There is a slight disagreement concerning propagation models in related work. Works [15, 17, 21, 24, 26, 27] apply conventional Ground-to-Ground (G2G) propagation models to Air-to-Ground (A2G) channel. Authors of [9, 12, 14, 16, 18–20, 22, 23, 25, 28] apply the A2G propagation model proposed in [9], while authors of [13] apply the model from [29]. In [32] authors propose the model, based on ray-tracing simulation results. The main difference between them is the way they calculate the total path loss considering different propagation segments in A2G channel. State-of-the-art A2G propagation models for low-altitude platforms have not yet been validated in real experiments.

We discuss the difference between G2G and A2G channels and compare state-of-the-art A2G models to real measurements obtained using the developed ABS prototype. We show that the A2G propagation model from [9] has better fit to experimentally obtained data in comparison to A2G models proposed in [29] and [32].

2.2 System Model and Assumptions

Before discussing the ABS placement problem, we need to introduce the system model. A summary of the model assumptions is given in Fig. 2.2. In this thesis all ABSs are assumed to be Long-Term Evolution (LTE)-compliant and for simplicity, only downlink channels are considered.

2.2.1 Capacity Estimation

The maximum communication capacity of a band-limited Additive White Gaussian Noise (AWGN) channel in bits per second can be estimated using the following equation [33]:

$$C = W \log_2(1 + SINR), \tag{2.1}$$

where W is the total channel bandwidth and $SINR$ is the Signal-to-Interference-and-Noise Ratio (SINR). The average SINR in turn is defined as:

$$SINR = \frac{S}{N + \sum_{i=1}^{n} I_i}, \tag{2.2}$$

where S is the received power of interest, N is the thermal noise power at the receiver side, I is the received power of i-th interfering signal and n is the number of active interfering transmitters in a given moment of time. The interference here is modeled as Gaussian noise. The thermal noise power at the receiver side is defined as:

$$N = K_B T_k W, \tag{2.3}$$

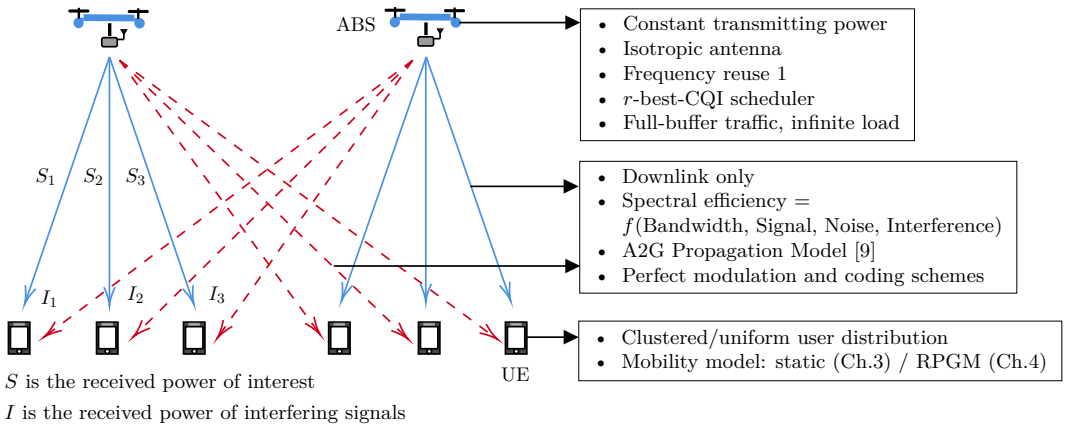


Figure 2.2. – Assumptions for different modeling aspects

where K_B is the Boltzmann constant, T_k is the temperature in Kelvin. Depending on the $SINR$, LTE system defines the Modulation and Coding Schemes (MCS) to increase the spectral efficiency and at the same time decrease the probability of link errors according to channel conditions [34]. We assume the availability of perfect MCS. The minimum SINR requirement is defined as:

$$SINR \geq \zeta_{th} \tag{2.4}$$

The minimum SINR requirement ζ_{th} depends on the network geometry and antenna configuration and is in practice determined from network measurements [35]. For simplicity, we assume that $\zeta_{th} = 1$. Thus, the maximum communication capacity can be defined as:

$$C(SINR) = \begin{cases} W \log_2(1 + SINR), & \text{if } SINR \geq \zeta_{th} \\ 0, & \text{if } SINR < \zeta_{th} \end{cases} \tag{2.5}$$

The SINR ratio depends on the power of the received signal of interest and the power of interfering signals. Figure 2.2 shows an example scenario for two ABSs sharing the same spectrum resource. As we can see, the SINR values at UEs degrade due to interference generated by the non-serving ABS.

2.2.2 Propagation Model

In a wireless radio channel, the received signal power is subject to the variation over distance due to path loss [33]. The received signal power can be expressed in dBm as:

$$S = P_t + 10 \log_{10} G_t + 10 \log_{10} G_r - PL \tag{2.6}$$

where P_t is the transmitted power in dBm, G_t is the gain of a transmitter antenna, G_r is the gain of a receiver antenna and PL is the path loss in dB. In this thesis we focus on the aspects of ABS placement and for simplicity we assume that P_t is constant and transmitter and receiver antennas are isotropic ($G_t = G_r = 1$).

Obstacles and reflecting sources in the vicinity of the UE have a significant influence on the characteristics of the propagation path. To estimate the path loss, accurate propagation models are required. Consider the A2G propagation path as shown in Fig. 2.3. The A2G propagation path can be divided into two: LoS and near-ground segments. Depending on the relative location of users to the ABS, the near-ground segment may be more or less obstructed. The obstruction is defined through the given environment and an *elevation angle* θ [36]. The elevation angle θ is defined as the

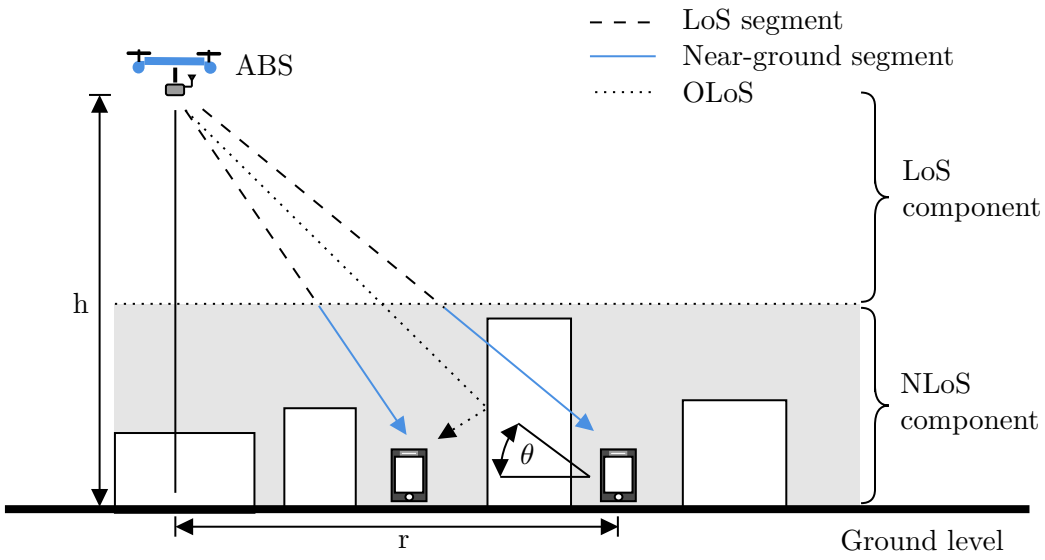


Figure 2.3. – Air-to-ground radio propagation environment.

acute angle in the range of 0° and 90°. The probability of the near-ground segment obstruction decreases as the elevation angle θ increases and decrease otherwise.

Recognizing the difference between the A2G and G2G channels, three A2G models have recently been proposed in the literature. Before describing them in the next section, we first introduce the log-distance path loss model, which forms a necessary basis for two A2G models.

2.2.2.1 Log-Distance Path Loss

The log-distance path loss model is used to predict the path loss for different environments. In the far field region of the transmitter, the path loss in dB is defined as follows:

$$PL = PL_{d_0} + 10 \cdot \gamma \cdot \log_{10}(d/d_0), \tag{2.7}$$

where PL_{d_0} is a constant which depends on the antenna characteristics, d_0 is a reference distance for the antenna far-field, d is the distance between a transmitter and a receiver in meters ($d > d_0$) and γ is the path loss exponent. The path loss exponent γ is given for different environments, e.g. free-space area, suburban area, urban area, dense urban area [33]. For the Free Space Path Loss (FSPL) the $\gamma = 2$.

2.2.2.2 Air-to-Ground Propagation Models

The comparison between state-of-the-art A2G path loss models is presented in Table 2.2. The main difference between them is the way they calculate the total path loss considering

different propagation segments in A2G channel. Next, we describe each model in detail.

Analytical A2G Path Loss Model [9]

Al-Hourani et al. proposed an analytical A2G path loss model in [9]. This model assumes two dominant propagation groups – LoS and Non-Line-of-Sight (NLoS) and uses the recommendation of the International Telecommunication Union (ITU-T) to calculate the LoS probability depending on the city parameters. In the ITU-T recommendation [37], the environment parameters are defined as:

- α_0: the ratio of land area covered by buildings to the total land area (dimensionless);
- β_0: the mean number of buildings per unit area (buildings/km^2);
- γ_0: the parameter that characterizes the building height distribution according to Rayleigh probability density function: $f(H) = (H/{\gamma_0}^2)exp(-H^2/2{\gamma_0}^2))$, where H is the building height in meters.

It is not always possible to accurately estimate these parameters and thus, the authors of the model have suggested approximating the ITU-T model and calculating the LoS probability using only two parameters as follows:

$$p_{LoS}(\theta) = \frac{1}{1 + a \cdot exp(-b[\theta - a])}, \tag{2.8}$$

where parameters a and b are city-specific Sigmoid curve parameters (from [9]) and θ is the elevation angle. The probability of NLoS is defined as:

$$p_{NLoS}(\theta) = 1 - p_{LoS}(\theta) \tag{2.9}$$

Table 2.2. – Comparison of the state-of-the-art A2G path loss models.

Model / Characteristic	**Model in [9]**	**Model in [29]**	**Model in [32]**
Scenario	suburban, urban, dense-urban, highrise-urban	suburban, urban, dense urban, highrise urban	dense urban (Bristol, UK)
Derivation Method	analytical	ray-tracing simulation	ray-tracing simulation
Frequency, GHz	n/a	0.7, 2, 5.8	1, 2, 2.5, 5
Total Pathloss	weighted sum of LoS and NLoS	either LoS or NLoS	sum of LoS, OLoS and NLoS

The individual path losses PL_{LoS} and PL_{NLoS} are calculated using the log-distance model (Eq. 2.7). For the LoS component PL_{LoS} the path-loss exponent γ is equal to 2 (FSPL), while for the NLoS component PL_{NLoS} is determined by environment type. Both propagation groups LoS and NLoS contribute to the total path loss. The total path loss is given by the weighted sum of both propagation groups:

$$PL = p_{LoS}(\theta) \cdot PL_{LoS} + (1 - p_{LoS}(\theta)) \cdot PL_{NLoS} \tag{2.10}$$

Segmented Excessive Path Loss Model [29]

Similar to the previous model, the A2G propagation model in [29] distinguishes LoS and NLoS propagation groups. However, only one of them contributes to the total path loss, depending on the probability of group occurrence. The probability of the LoS propagation group to occur with respect to the elevation angle θ is expressed as follows:

$$p_{LoS}(\theta) = c_\xi(\theta - \theta_0)^{d_\xi}, \tag{2.11}$$

where θ_0 is 15° (the minimum elevation angle for the model), c_ξ and d_ξ are the frequency and the environment dependent parameters obtained from the curve fitting, which can be found in [29]. The probability of the NLoS propagation group occurrence is given as follows:

$$p_{NLoS}(\theta) = 1 - p_{LoS}(\theta), \tag{2.12}$$

To decide which propagation group to use, a random number R is generated between 0 and 1 and the following rules are executed:

- if $p_{LoS}(\theta) \geq p_{NLoS}(\theta)$, then if $R \geq p_{NLoS}(\theta)$: pick the LoS group, otherwise the second group;
- if $p_{NLoS}(\theta) \geq p_{LoS}(\theta)$, then if $R \geq p_{LoS}(\theta)$: pick the NLoS group, otherwise the first group.

Finally, the path loss can be expressed as:

$$PL = PL_{LoS} + \mathcal{N}(\mu_\xi, \sigma_\xi^2(\theta)), \tag{2.13}$$

where PL_{LoS} is the FSPL from Eq. 2.7, $\mathcal{N}$ is the normal distribution with mean μ and standard deviation $\sigma(\theta)$ and ξ representing the propagation group. It is assumed that the path loss mean μ is constant and the standard deviation $\sigma(\theta)$ is a function of the elevation angle θ. The standard deviation is given as follows:

$$\sigma_\xi(\theta) = a_\xi exp(-b_\xi \theta), \tag{2.14}$$

where a_ξ and b_ξ are the frequency and the environment-dependent parameters which can be found in [29].

Three-component A2G path loss model [32]

Feng et al. [32] performed extensive ray-tracing simulations for different ABS altitudes (100m, 200m, 500m, 1000m and 2000m) and carrier frequencies (200MHz, 1GHz, 2GHz, 2.5GHz and 5GHz). The simulation environment represents Bristol City in the United Kingdom as an example of a typical urban European city. They proposed distinguishing between three propagation components: the LoS, the Obstructed-Line-of-Sight (OLoS) and the NLoS components as shown in Fig. 2.3.

In this model, the path loss for the LoS component is defined as:

$$PL_{LoS}(\theta) = -20 \cdot \log_{10} sin(\theta), \tag{2.15}$$

The path loss for the OLoS component is defined as:

$$PL_{OLoS}(\theta) = \eta_0 - \eta_1 e^{(90-\theta)/v_0}, \tag{2.16}$$

where the coefficients η_0, η_1 and v_0 are the best fit parameters to the simulation data and can be found in [32]. The path loss for the NLoS component is defined as:

$$PL_{NLoS}(\theta) = \eta_2 - \eta_3 e^{(90-\theta)/v_1}, \tag{2.17}$$

where coefficients η_2, η_3 and v_1 are the best fit parameters to simulation data and can be found in [32]. Finally, the total path loss is calculated as follows:

$$PL(\theta) = PL_{LoS}(\theta) + PL_{OLoS}(\theta) + PL_{NLoS}(\theta), \tag{2.18}$$

where $\theta > 10^\circ$ (the minimal elevation angle for this model).

2.2.2.3 Summary

As may be seen, the state-of-the-art A2G propagation models separate the propagation path into several components and model their individual occurance probabilities. However, probability prediction and path loss estimation methods differ for different models. To choose a suitable model to be used in this thesis, we have conducted the real-world measurements described in Chapter 5. The collected experimental data was compared to the data generated by state-of-the-art models. The model proposed in [9] shows good prediction properties, while the two other models show significantly lower

correlation with the experimental data. The generated Received Signal Strength (RSS) data for different elevation angles θ are shown in Fig. 2.4.

Thus, we use the model from [9] for the analysis and simulations in this thesis. Using the propagation model together with the equations from Section 2.2.1, the capacity for the A2G channel can be estimated.

2.2.3 Resource Management

In LTE, to exploit the variations in the instantaneous radio channel conditions, the channel-quality dependent scheduling is used. The goal is to share the radio resource between multiple UEs efficiently. In this work, we discuss only downlink communication and assume a constant transmission power at ABSs.

2.2.3.1 Radio Resource Allocation

In LTE, Orthogonal Frequency-Division Multiplexing (OFDM) is used, which allows a radio scheduler to exploit the channel variations in time and frequency domain together. The smallest element of resource allocation which can be assigned by the BS scheduler is a Resource Block (RB). RB is defined as consisting of 12 consecutive subcarriers for one slot (0.5 ms) in duration as represented by the resource grid in Fig. 2.5.

RBs can be allocated in time, as well as in the frequency domains. In this way, the system can provide greater operational flexibility and the scheduler can assign a different amount of RBs upon the UE requirements. Also, the physical layer specifications are

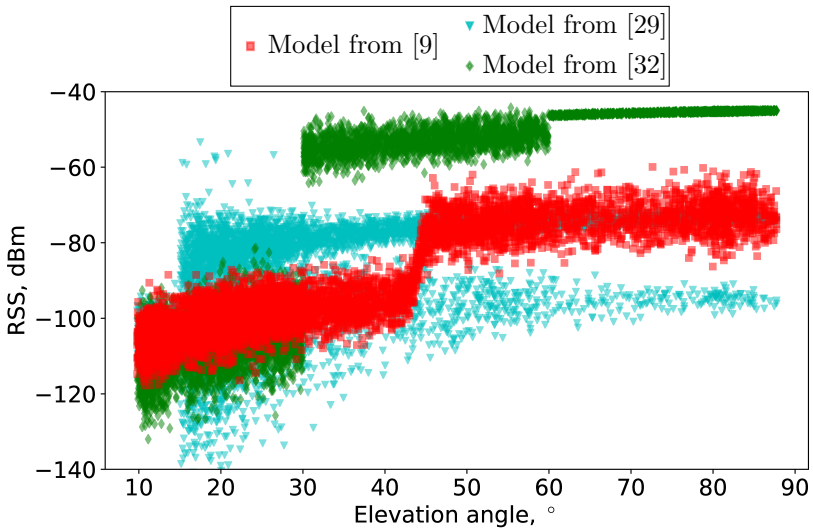


Figure 2.4. – Simulated RSS vs. elevation angle θ for state-of-the-art A2G propagation models in urban scenario.

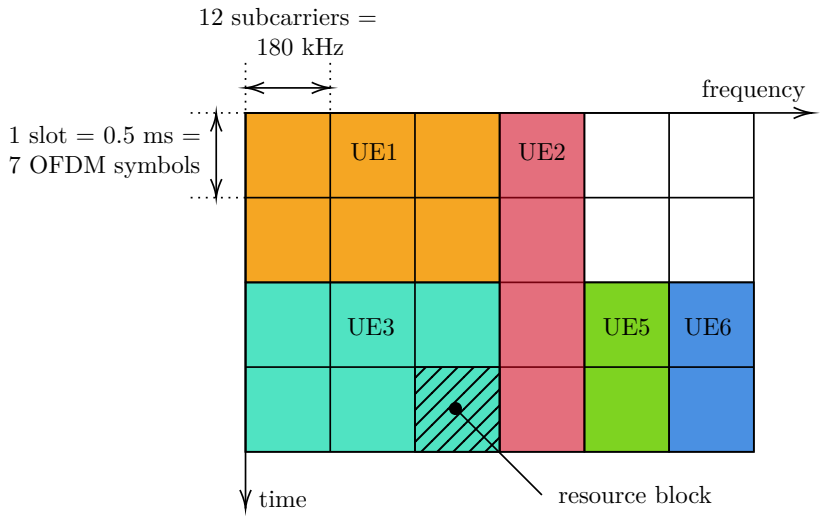


Figure 2.5. – Simplified downlink resource grid in LTE. Each box within the grid represents a resource block.

not dependent on bandwidth and designed to accommodate up to 20 MHz of system bandwidth so that the total number of RBs depends on the overall system transmission bandwidth as shown in Table 2.3.

2.2.3.2 Traffic Behavior and Scheduling Strategy

We assume that the network experiences traffic overload and applies a full buffer traffic model for each user. As we consider a downlink only, each user always has data to receive. It is worth noting that if some users do not have information to receive, they can be removed from the placement optimization problem description without any loss of generality.

The distribution of resources to users depends on the applied radio scheduler [39]. The scheduler can increase the system spectral efficiency by assigning resources to the users that can utilize them best. Scheduling algorithm design is a big research area and there are many schedulers proposed [39, 40].

Table 2.3. – Orthogonal Frequency Division Multiple Access (OFDMA) parameters for LTE E-UTRA [38].

Bandwidth (MHz)	1.25	2.5	5.0	10.0	15.0	20.0
Subcarrier bandwidth (kHz)	15					
Subcarrier duration (ms)	1.0					
RB bandwidth (kHz)	180					
Number of available RBs	6	12	25	50	75	100

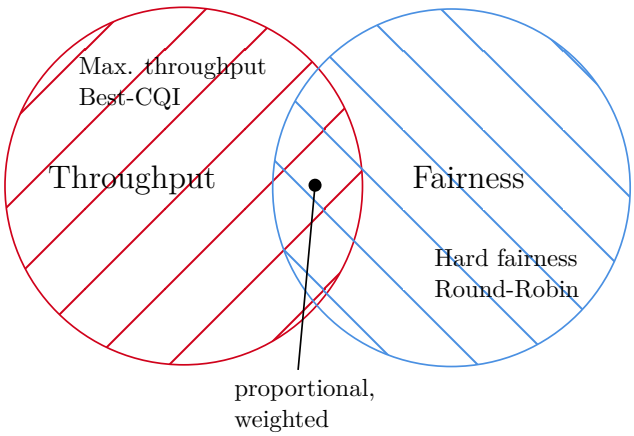


Figure 2.6. – Main types of radio schedulers and their objectives.

However, the choice of the algorithm depends on the considered scenario. In [41] we have investigated the influence of different schedulers on the performance of multiple ABS placement strategies in terms of the aggregated system capacity and the scheduling fairness. Our results show that the objective of the scheduler (e.g. throughput or fairness) should match the objective of the placement algorithm (e.g. traffic offloading or user coverage). There are two basic scheduler types which have opposite objectives as shown in Fig. 2.6 [42]:

- Round Robin (RR) Scheduler: it assigns resources in turn (one after another) without taking channel conditions into account. This is a fair scheduling scheme since every user is given the same amount of resources. However, the overall spectral efficiency and throughput are generally much lower than in other schedulers, because users with good channel conditions can not take their advantage.

- Best-CQI: assigns resources to UEs with the best channel conditions based on Channel Quality Indicator (CQI) indicator. UEs located far away from the base station are unlikely to be scheduled. This scheme can increase cell throughput at the expense of fairness.

We adapt the r-best-CQI scheduler, which assigns resources to r UEs with the best channel conditions, where r is the number of available RBs for the ABS. In Chapter 3 we compare RR and r-best-CQI schedulers and show the impact of resource scheduling strategy on aggregated capacity.

Since we consider the network overload situation, we ensure that the number of UEs is always greater than the number of available RBs multiplied by the number of ABSs. Hence, there is always demand for radio resources (e.g. infinite load).

2.2.3.3 Aggregated System Capacity

Aggregated system capacity shows the overall throughput of the system. The user throughput, C, defined as the maximum transmission data rate that the user can achieve on a given channel depends on the user's SINR as in Eq. 2.5. The total system capacity or aggregated system capacity can be calculated as follows:

$$C_{total} = \sum_{i=1}^{m} \sum_{j=1}^{r} C_{ij}, \tag{2.19}$$

where m is the number of ABSs in the system, r is the number of available RBs for each ABS and C_{ij} is the user throughput connected to i ABS and assigned with j RB as in Eq. 2.5.

2.2.4 User Distribution and Mobility Modeling

The aerial base station placement and its performance depend on user distribution. In the next sections, we describe the user distribution and the mobility of UEs and ABSs. For Chapter 3 we assume to be given a snapshot of the network with a certain user distribution, which we use in the ABS placement optimization problem. For Chapter 4 we assume moving ABSs and include the user mobility.

2.2.4.1 User Distribution

Users can be uniformly distributed among the area or form groups, which we call *clusters*. In related work, it is usually assumed that the user distribution is either uniform or is given by a predefined function [43].

For reasons of generalization, we have developed a generic parametric function to create various user distributions. This function allows a smooth transition between highly clustered and uniform distributions. Each cluster is defined by three parameters as shown in Fig. 2.7:

- *cluster center* Z or *centroid*, which corresponds to the mean of the user locations in a cluster;
- *number of clusters in one dimension* q; q^2 gives a number of clusters in 2D space;
- *normalized sparsity* α defines the normalized deviation of the user positions from the centroid Z. The $\alpha = 1$ corresponds to a uniform user distribution, while $\alpha = 0.1$ results in q^2 dense clusters.

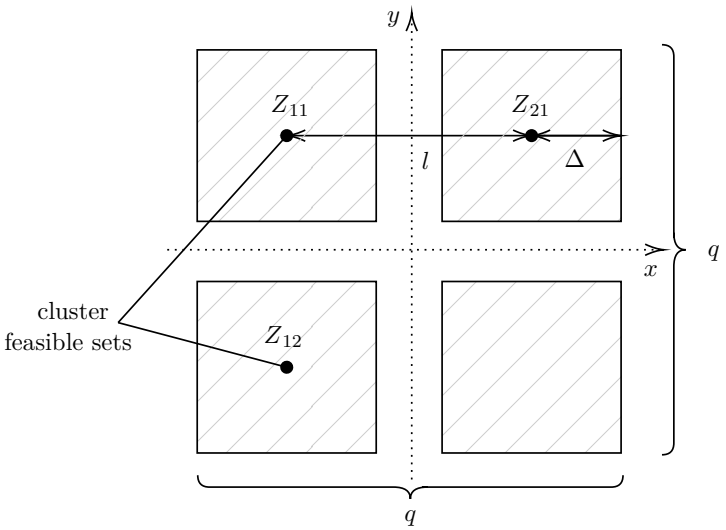


Figure 2.7. – Cluster generation parameters.

Centroid coordinates for clusters in the target area are defined as follows:

$$Z = \begin{bmatrix} (l,l) & (2l,l) & \cdots & (ql,l) \\ (l,2l) & (2l,2l) & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ (l,ql) & \cdots & \cdots & (ql,ql) \end{bmatrix}, \tag{2.20}$$

where l is the distance among cluster centroids in one dimension. It is defined as follows:

$$l = \frac{B}{q+1}, \tag{2.21}$$

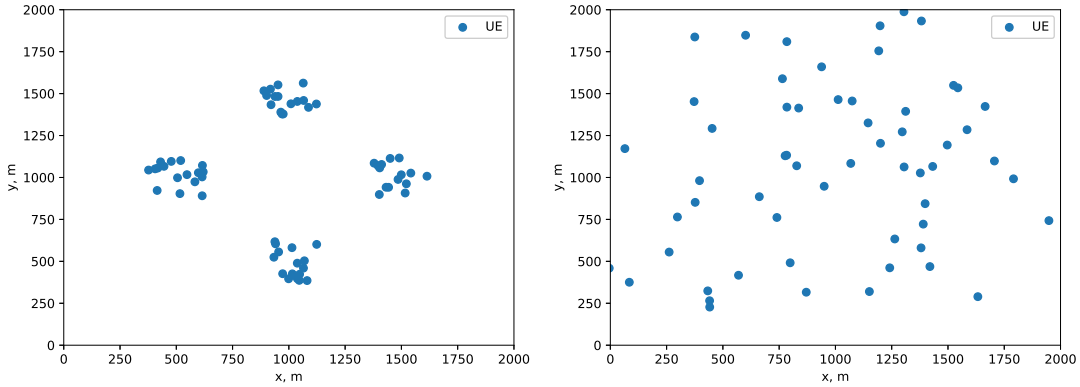
where B is the side length of the square target area.

The absolute deviation Δ from the centroid Z (which corresponds to the distance among clusters in one dimension) is calculated as follows:

$$\Delta = \alpha \cdot l \tag{2.22}$$

When the sparsity α is low, the dense user clusters are generated as shown in Fig. 2.8(a). When $\alpha = 1.0$, users form a uniform-like distribution as shown in Fig. 2.8(b). To prove that generated distributions with $\alpha = 1.0$ are indistinguishable from the uniform distribution, the two-sample Kolmogorov–Smirnov (KS) test was used [44].

KS test quantifies the distance between the empirical distribution functions of two samples and tries to reject the null hypothesis that the samples are taken from the same



(a) High cluster density (clustered, $\alpha = 0.1$) (b) Low cluster density (uniform-like, $\alpha = 1.0$)

Figure 2.8. – Examples of clustered and uniform-like distributions for 64 users, $q^2 = 4$.

distribution. The KS metric is expressed as follows:

$$D_{n,m} = \max_x |F_{1,n}(x) - F_{2,m}(x)|, \tag{2.23}$$

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of the first and the second samples, n and m are the sizes of the first and the second samples, max is the upper bound function. The empirical distribution function gives the proportion of the data that lies below a certain value.

The p-value or probability value is the probability that, when the null hypothesis is true, the mean difference between two samples would be equal to, or more extreme than, the actual observed results. The p-values for certain values of the KS metric and the number of samples can be found in [45].

For this test we have generated 1000 distributions of 64 users using the developed function ($\alpha = 1.0$) and the uniform distribution generator. The null hypothesis is that these two distributions are generated from the same source. To perform the KS test, we have used the function *scipy.stats.ks_2samp* from the open source scientific framework *SciPy* [46]. A comparison of these distributions showed that the average value of KS metric is $D_{n,m} = 0.14$, while the probability value $p = 0.24$. It is not possible to reject the null hypothesis, since the p-value $0.24 > 0.05$, where 0.05 is the level of statistical significance, and $D_{n,m} = 0.14$. This means that the maximum difference between the two distributions is relatively small. Strictly speaking, it is not possible *to prove* that our function generates uniformly distributed users for $\alpha = 1.0$, but we can claim that our function generates a distribution that is *not distinguishable* from uniform distribution.

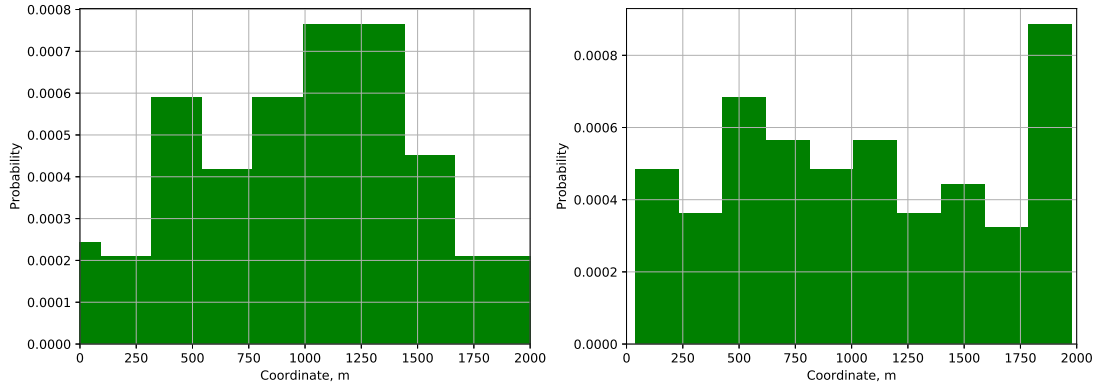
In Fig. 2.9 two randomly generated datasets from different distributions are compared. It can be seen that the differences between them are minor.

The developed function provides a simple and reliable way to control the distribution of users. In this thesis, the user distribution model defines the initial user locations, while the mobility model modifies the user distribution at runtime if the users are not static.

2.2.4.2 User Mobility

In the real world, people often form groups and move together in a certain direction. For example, during the sports event, athletes usually move as a group to a destination, while their fans move together to get a better overview of the event. Thus, there are often some spatial dependencies among users. To model them, we use the Reference Point Group Mobility Model (RPGMM) model [47].

In this model, each user belongs to a group and follows the logical center of the group (leader). The leader defines the groups' motion behavior. It is assumed that the group leader follows the random waypoint model. In the random waypoint model, the waypoints are uniformly distributed over the area. After reaching each waypoint, the group leader chooses a new velocity uniformly at random in an interval (v_{min}, v_{max}) and continues to the next waypoint [48]. Each group member adds a random deviation to the waypoint of the group leader, which is uniformly distributed in the interval $[0, r_{max}]$, where r_{max} is the maximum allowed deviation.



(a) Uniform-like ($\alpha = 1.0$)

(b) Uniform

Figure 2.9. – Histograms of the two generated distributions for 64 users, $q^2 = 4$.

2.2.4.3 ABS Mobility

In this thesis, we select the simple model in which the aircraft is modeled as a particle with a given acceleration and maximum speed [49]. We make the following assumptions: (1) the aircraft is rigid and does not experience any deformations, (2) the Earth is flat and non-rotating, the gravity acceleration is constant and perpendicular to the Earth surface, (3) the atmosphere properties are constant. To describe the ABS motion effectively, we use two reference frames: the global frame, which is fixed to the surface of Earth, and the body frame, which is fixed to the aircraft. In the body frame, the current velocity vector is aligned with y-axis of the aircraft as shown in Fig. 2.10.

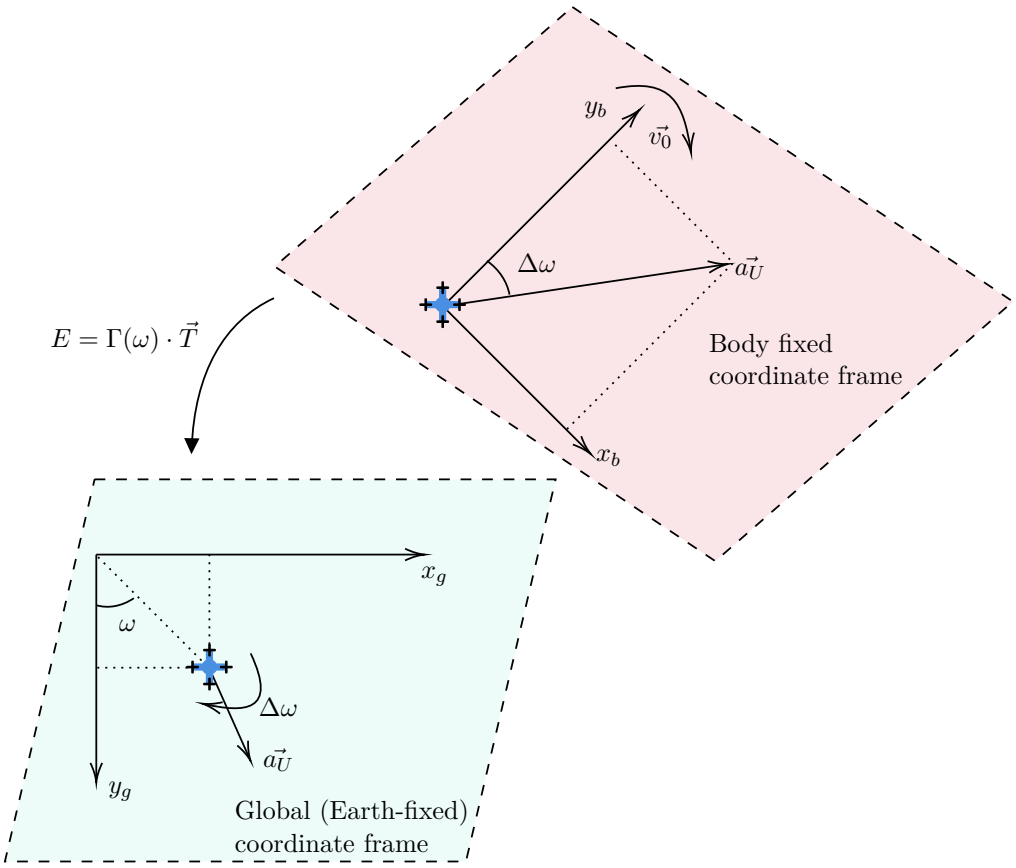


Figure 2.10. – The aircraft in the global coordinate frame and the body-fixed coordinate frame. The horizontal and angular displacements are calculated in the body frame and then using rotation and translation operations are converted into the global frame to find the new aircraft position.

The body frame is used to calculate the horizontal displacement $(\Delta T_x, \Delta T_y)$ and angular displacement $\Delta\omega$ of the ABS:

$$\begin{aligned} \Delta T_x &= (v_0 t + \frac{1}{2} a_U t^2) \cdot \cos \Delta\omega, \\ \Delta T_y &= (v_0 t + \frac{1}{2} a_U t^2) \cdot \cos \Delta\omega, \\ \Delta\omega &= v_r \cdot t, \end{aligned} \tag{2.24}$$

where t is the elapsed time, v_0 is the previous velocity, a_U is the acceleration of the vehicle, v_r is the angular velocity and Δt is the elapsed time.

To obtain the new position of the aircraft in the global coordinate frame, the obtained displacement $(\Delta T_x, \Delta T_y)$ and $\Delta\omega$ in the body frame are converted using a rotation and a translation operations. The rotation matrix is defined as follows:

$$\Gamma(\omega) = \begin{bmatrix} cos(\omega + \Delta\omega) & -sin(\omega + \Delta\omega) & 0 \\ sin(\omega + \Delta\omega) & cos(\omega + \Delta\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.25}$$

where ω is the bearing angle (the horizontal angle between the current direction of the aircraft and y-axis in the global frame coordinates. The translation vector is defined as follows:

$$\vec{T} = \begin{bmatrix} \Delta T_x \\ \Delta T_y \\ 1 \end{bmatrix}, \tag{2.26}$$

where $(\Delta T_x, \Delta T_y)$ is the horizontal displacement of the aircraft. Then, the rotation, followed by translation, is as follows:

$$E = \Gamma(\omega) \cdot \vec{T} = \begin{bmatrix} cos(\omega + \Delta\omega) & -sin(\omega + \Delta\omega) & \Delta T_x \\ sin(\omega + \Delta\omega) & cos(\omega + \Delta\omega) & \Delta T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

The new position of the aircraft (T_x, T_y) is as follows:

$$\begin{bmatrix} T_x \\ T_y \\ 1 \end{bmatrix} = E \cdot \begin{bmatrix} T_{x0} \\ T_{y0} \\ 1 \end{bmatrix}, \tag{2.28}$$

where (T_{x0}, T_{y0}) are the previous aircraft coordinates.

We also assume that there are no external disturbance components in the system, such as wind or obstacles.

2.3 Simulation Method and Constant Simulation Parameters

We use the Monte Carlo method for simulation, relying on repeated random sampling to obtain numerical results [50]. For each of the simulation scenarios, at least 64 runs are processed and the results are averaged. Constant simulation parameters are summarized in Table 2.4.

2.4 Conclusion

The related work shows great potential in the use of ABS systems for traffic offloading during the overload of existing cellular infrastructure. One of the most important problems is the ABS placement optimization problem in the dynamic temporary-event scenarios. It is computationally complex and requires fast and accurate algorithms for ABS placement. Another important issue is the modeling of A2G radio channel. State-of-the-art works use different channel models, and so far these models have not been experimentally validated. In this chapter, we have briefly described important modeling aspects and assumptions made during modeling.

Table 2.4. – Constant simulation parameters.

Parameter	**Value/Name**
Carrier frequency	1800 MHz
System bandwidth	1.25 MHz, 6 RBs
Propagation model	A2G [9], ($a = 46$, $b = -2$), ($\gamma = 2.5$, $\sigma = 7.0$)
ABS transmit power	20 dBm
ABS and UE antenna gains	1
Frequency reuse, ABSs	$FR = 1$
ABS scheduler	r-best-CQI ($r = 6$)
Traffic model	Full buffer
Area size	2000x2000 m^2
RPGMM user mobility	$v_{min} = 0.5$ m/s, $v_{max} = 2$ m/s, $r_{max} = \alpha$
ABS maximum speed	4 m/s
ABS acceleration	1 m/s^2
ABS angular velocity	2 rad/s
Monte-Carlo simulation runs	64

Chapter 3

Centralized Aerial Base Station Placement for Static UEs

Man errs as long as he doth strive.

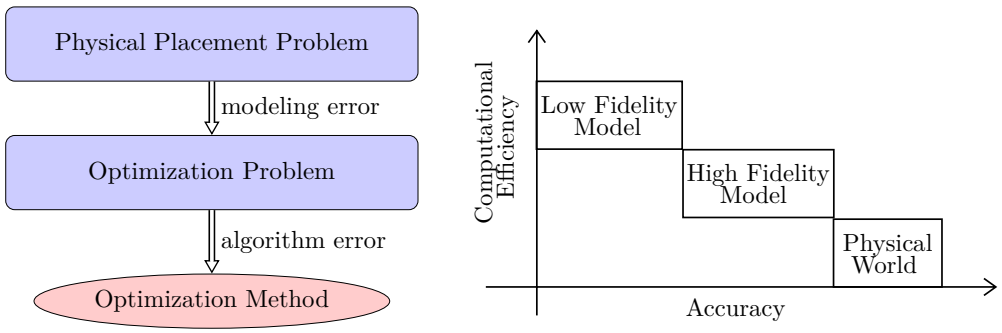
Johann Wolfgang von Goethe

In this chapter we discuss centralized algorithms for the ABS placement, which aim to maximize the aggregated system capacity for traffic offloading during temporary-events. We assume that all ABSs reuse the frequency resource in space. As described in the previous chapter, in this case, the capacity is estimated using an SINR-based capacity model. In the following sections we formulate the optimization problem of finding ABSs locations, which maximize the aggregated system capacity, and show that the problem is NP-hard.

Due to the dynamic nature of temporary-events, it is necessary to update solutions for the ABS placement optimization problem. Thus, computationally-efficient heuristics need to be developed in order to recalculate the ABS placement frequently.

Considering the problem-specific knowledge, we approximate the placement optimization problem by a fast distance-based capacity estimation function, for which efficient heuristics exist. Unfortunately, as shown in Fig. 3.1(b) the problem approximation introduces a certain error. In order to ensure the solution quality, we propose a novel algorithm, which (1) prepares a set of feasible solutions based on clustering, (2) evaluates them using the accurate SINR-based capacity calculation and selects the best solution.

The solution of the physical placement problem is affected by two types of errors, as shown in Fig. 3.1(a): (1) modeling error, (2) algorithm error. Modeling a physical placement problem always introduces a certain error. A good model should reflect factors that significantly affect the physical problem and abstract factors that have no



(a) Optimization error (b) Model performance

Figure 3.1. – Conceptual view on optimization problem and modeling error.

significant impact. After the appropriate model is chosen, the optimization problem can be formulated. In this thesis we mainly consider heuristic optimization, which does not guarantee solution optimality. Each algorithm introduces a certain error, which depends on the performance of the algorithm. There is an error that depends on the performance of the algorithm. We discuss the impact of modeling and algorithm errors on a physical placement problem in the following sections.

3.1 SINR-based Capacity Optimization

The centralized ABS placement problem is given as follows: given user locations in a particular moment of time, the near-optimal ABS locations under the respective resource allocation scheme need to be found in order to support the maximum offloading capacity. As shown in Fig. 3.2, in the optimization problem the signal power between ABS and targeted UEs, as well as the interference signal power from neighboring ABSs on downlink, need to be considered.

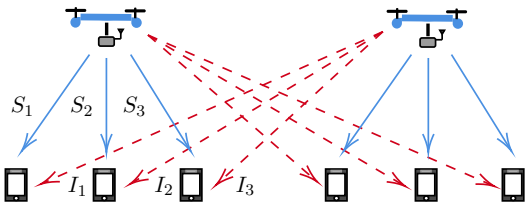


Figure 3.2. – Example scenario for two ABSs, sharing the same spectrum resource. The downlink interference created by the non-serving ABS degrades the SINR at UEs.

3.1.1 Problem Definition

Let s be the set of the ABS locations and u be the set of the user locations:

$$\begin{aligned} s &= (s_1, \ldots, s_m), \text{where } s_j = (x_j, y_j, z_j), \forall j \in [1, m], \\ u &= (u_1, \ldots, u_n), \text{where } u_i = (x_i, y_i, z_i), \forall i \in [1, n], \end{aligned} \tag{3.1}$$

where x, y, z are coordinates, m is the number of ABSs, n is the number of users, $s \in \mathbb{R}^{3m}$ and $u \in \mathbb{R}^{3n}$.

The distance between i-user and j-ABS is calculated as follows:

$$d_{ij} = \|u_i - s_j\| \tag{3.2}$$

Then, the problem of maximization of the aggregated capacity for ABSs is expressed as:

$$\begin{aligned} \underset{s,R}{\text{argmax}} \quad & \sum_{j \in s} \sum_{i \in u} R_{ij} \cdot C_{ij} \\ \text{subject to} \quad & R_{ij} \in \{0, 1\}, \\ & \sum_{i=1}^{n} \sum_{j=1}^{m} R_{ij} = r \cdot m, \end{aligned} \tag{3.3}$$

where R_{ij} is equal to 1 if the i-th UE is serviced by j-th ABS and it is 0 otherwise, r is the number of available RBs for each ABS.

The capacity C_{ij} is calculated as follows:

$$\begin{aligned} C_{ij} = \quad & W \log_2 \left(1 + \frac{S(d_{ij})}{N + \sum_{l \in s, l \neq i} R_{il} \cdot I(d_{il})} \right), \\ \text{subject to} \quad & S(d_{ij}) / (N + \sum_{l \in s, l \neq i} R_{il} \cdot I(d_{il})) \geq \zeta_{th}, \\ & R_{il} \in \{0, 1\}, \end{aligned} \tag{3.4}$$

where R_{il} is equal to 1 if the i-th UE is experiencing interference from l-th ABS and if not it is equal to 0, W is the channel bandwidth, S is the received power at the user side, N is the power of thermal noise, I_l is the received power of interfering signals from the l-th ABS and ζ_{th} is the minimum SINR requirement.

The received power is estimated using the A2G propagation model as given in Section 2.2.2.

Substituting Eq. 3.4 in Eq. 3.3, we obtain the optimization problem as follows:

$$\begin{aligned} \underset{s,R}{\text{argmax}} \quad & \sum_{j\in s}\sum_{i\in u} R_{ij}\cdot W\log_2\left(1+\frac{S(d_{ij})}{N+\sum_{l\in s,l\neq i} R_{il}\cdot I(d_{il})}\right), \\ \text{subject to} \quad & R_{ij}\in\{0,1\}, \\ & R_{il}\in\{0,1\}, \\ & \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij} = r\cdot m, \\ & S(d_{ij})/(N+\sum_{l\in s,l\neq i} R_{il}\cdot I(d_{il}))\geq \zeta_{th}. \end{aligned} \tag{3.5}$$

3.1.2 Computational Complexity

NP-complete decision problem must satisfy two criteria: it must be in NP, and it must be Non-deterministic Polynomial-time hard (NP-hard), i.e. all problems in NP must be reducible to it. A proof of NP-completeness must prove both of these properties, while a proof of NP-hardness requires only the second [51, 52]. To prove these properties and show the NP-completeness of the defined placement problem, we use the set covering problem, one of the classical problems in combinatorics and computer science. The decision version of set covering is NP-complete, and the optimization/search version of set cover is NP-hard [53]. For our problem, the decision version can be expressed as follows:

Problem 3.1. *A set of UEs and their assignment to a set of ABSs are given. Is the aggregated system capacity greater than a specific value C_s?*

Lemma 3.2. *Decision problem 3.1 can be reduced to the set cover problem.*

Proof. Suppose all UEs are presented by a universe $\mathcal{U}$. UEs operated by a particular ABS can be represented as a family $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_n\}$. Families for multiple ABSs are subsets of $\mathcal{U}$. Thus, a cover is a subfamily $\mathcal{C} \subseteq \mathcal{S}$ of sets whose union is $\mathcal{U}$. Every subset $\mathcal{S}_i$ has an associated cost c, which is represented by aggregated system capacity.

In the set covering decision problem, the input is a pair $(\mathcal{U}, \mathcal{S})$ and every subset has an associated cost c; the question is whether there is a set covering with bigger or equal revenue than predefined value k. Since both problems are essentially the same and $k = C_s$, the original problem in 3.1 can be reduced to the set covering problem, which is NP-complete. □

Lemma 3.3. *Solution to the Problem 3.1 can be verified in deterministic polynomial time.*

Proof. Suppose there is a solution to the Problem 3.1, which needs to be evaluated e.g. ABS locations s. According to Eq. 3.4 to calculate the capacity for each UE, we need to calculate the received power of the serving ABS and the interference power of all interfering ABSs (with constant cost c_{SINR}). As shown in Fig. 3.3, the complexity is $O(rm + r(m-1)m) = O(rm^2)$, where r is the number of available RBs for each ABS and m is the number of ABSs. Then, the sum of capacity values for each UEs can be calculated in linear time. Thus, the function has a deterministic number of computational steps and thus the value of an objective function can be obtained in deterministic time. Now only the last step is necessary: a comparison of the predefined value C_s with the obtained value of the target function, which can be performed in deterministic time. Thus it can be concluded that a solution can be verified in deterministic polynomial time. □

Lemma 3.4. *The decision version of the problem is NP-hard and the exact solution can not be found in linear or sublinear time.*

Proof. The proof of NP-completeness is constructed by the problem reduction from the known problem in NP (see Lemma 3.2) and by the proof that a solution for this problem can be verified in deterministic polynomial time (see Lemma 3.3). Thus, the Problem 3.1 is NP-hard and the exact solution can not be found in linear or sublinear time. □

The decision version of set covering is NP-complete, and the optimization/search version of set cover is NP-hard. Thus, the optimization problem introduced in the

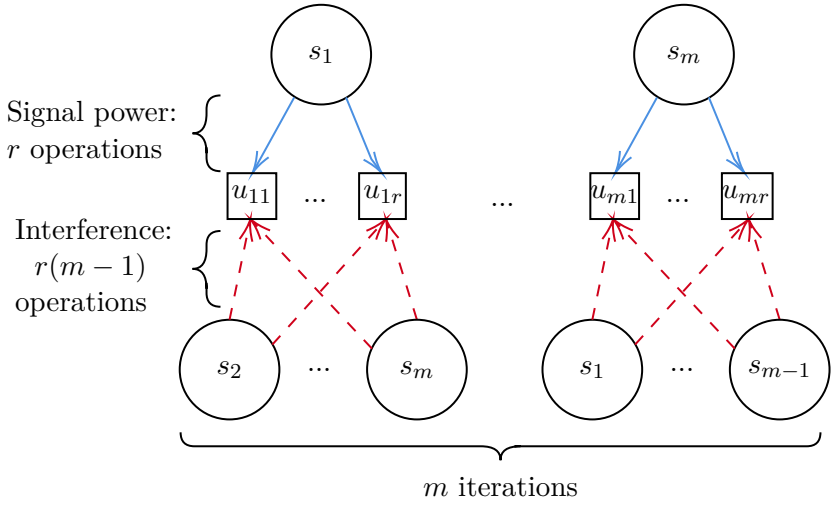


Figure 3.3. – The complexity of a solution evaluation on the SINR-based capacity model.

previous section is at least NP-hard. At this point, it is clear that the complexity of the problem increases exponentially with the size of the problem.

3.1.3 Properties of the Capacity Optimization Problem

Two capacity diagrams for two ABSs are shown in Fig. 3.4 and 3.5. These figures show the search space for the simplified optimization problem ($m = 2$), where the position of the first ABS is fixed at (500, 1000) and the position of the second ABS must be determined. It is assumed that each ABS has a fixed altitude of $z = 100$ m and $r = 6$ RBs. The x axis and y axis determine possible placement positions, while the color code and contours represent the system capacity in MBit/s. Fig. 3.4 shows the objective function for the uniform UE distribution, while Fig. 3.5 shows it for the 4 dense UE clusters.

Surface plots in Fig. 3.4 and 3.5 show that the search space of the simplified problem ($m = 2$) has more than one local maxima and thus is *not concave*. The convergence time and the quality of a solution can not be strictly guaranteed in this case [54].

The minimum SINR requirement ζ_{th} in Eq. 3.5 may result in discontinuity points in the search space. The function is *not smooth* at these points, and therefore the

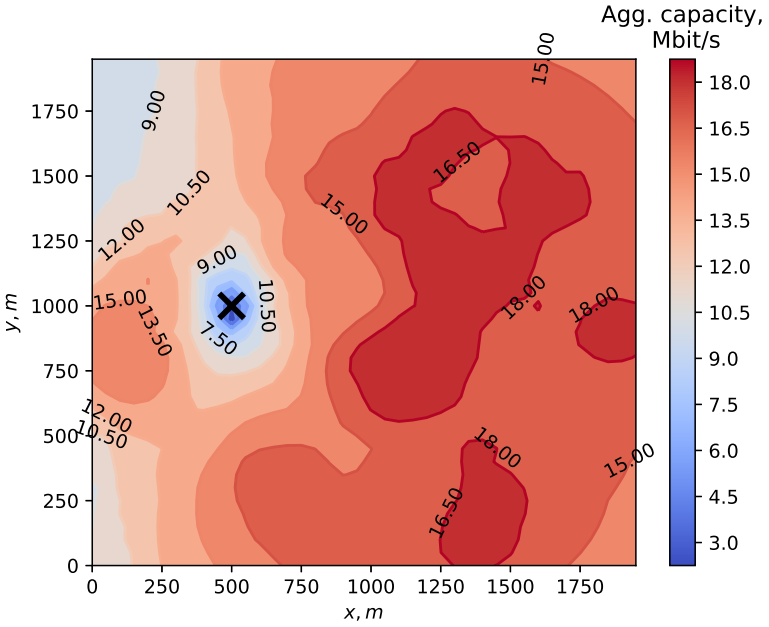


Figure 3.4. – Capacity plot of the system capacity for uniform distribution of UEs (6 RBs, 64 UEs, $\gamma = 2.5$, $\alpha = 1.0$). The first ABS is already fixed at the position (500, 1000), marked by the cross, and the position of the second ABS must be determined.

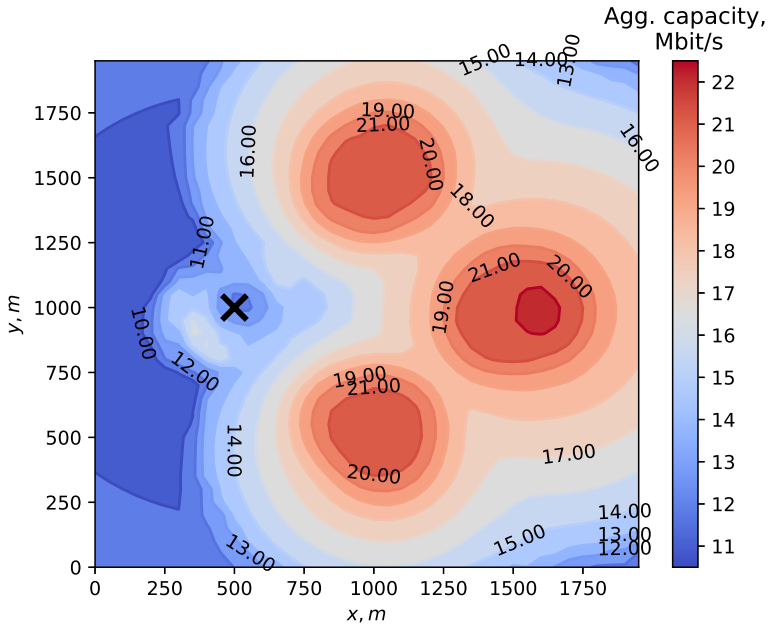


Figure 3.5. – Capacity plot of the system capacity for 4 dense UE clusters (6 RBs, 64 UEs, $\gamma = 2.5$, $\alpha = 0.1$). The first ABS is already fixed at the position (500, 1000), marked by the cross, and the position of the second ABS must be determined.

function slope can not be estimated. Many optimization algorithms use the gradient and therefore, without an additional regularization, can not be applied. The smoothness of the capacity varies in different regions and for different user distributions. This behavior is common for ill-posed optimization problems and indicates that solutions may suffer from *numerical instability* [51].

Considering these properties, this optimization problem can be classified as a complex, intractable problem and needs heuristic search algorithms to be solved in reasonable time. However, we can see certain *regularities* in the problem model, coming from the natural behavior of the propagation model and the fact that the signal power depends on the distance between a transmitter and a receiver:

- Decrease of the distance between the transmitter and the receiver increases the received signal power at the receiver: $S(d) \uparrow$ when $d \to 0$; an ABS should be located as close to UEs as possible to maximize the SINR ratio;

- Increase of the distance between the receiver and the interfering transmitter increases the SINR ratio: $I(d_I) \to 0$ when $d_I \to \infty$; UEs, served by the particular ABS, should be separated from interfering ABSs to minimize interference.

These properties can be used for problem approximation as a priori information about the problem.

3.1.4 State-of-the-Art Optimization Methods for the SINR-based Capacity Problem

The exact solution for the NP-hard problem can be obtained by an exhaustive search. However, with the increase of the problem size, the computational time grows exponentially and soon becomes too large for real applications, even for small problem sizes. In this case, heuristics can be used for problem solving, trading the accuracy of the solution for speed.

As we already know, certain algorithms are proposed in the literature to solve the problems of ABS placement as shown in Table 2.1 in Section 2.1. While there is no work considering the same optimization problem, we have adapted the closest approaches to solve the capacity optimization problem with multiple ABSs. In particular, there are slight differences in the related work regarding the following aspects:

- optimization objective (e.g. maximizing of number of covered users / minimizing latency instead of maximizing capacity);
- number of ABSs (e.g. consideration of a single ABS instead of multiple);
- resource reuse pattern (e.g. use of ortogonal frequency resources instead of frequency reuse 1);
- propagation models (e.g. conventional G2G models despite the utilization of A2G channel).

Several works propose to use PSO [19, 28]. It belongs to the class of stochastic globalized heuristics, which try to avoid local minima by performing multiple local searches in objective space. As suggested in multiple works [55, 56], we also consider the Nelder-Mead simplex algorithm as the often applied derivative-free heuristic for conventional BS placement. We excluded the method using a neural network [24] from consideration because the trained network needs a significant time for learning and in the case of changing the environment, it is necessary to retrain the neural network, which is associated with high data acquisition time and computing costs. We have also implemented a Random Search (RS) algorithm, which is used as a baseline for other approaches. These algorithms do not use problem-specific knowledge and can generally be applied to any problem.

3.1.4.1 Grid Search

To guarantee an optimal solution for a non-convex problem, it is necessary to employ an *exhaustive search*. It is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem statement.

On the continuous optimization problem, the exhaustive search requires infinitely many function evaluations. In this thesis, the placement region for the ABSs is bounded, and thus the problem can be approached by selecting a large number of grid points for possible ABSs locations. This approach is called *grid search*. The complexity of the grid search is p^m, where p is the number of grid points for every ABS. Thus, this method can be applied successfully only to small-sized problems or in combination with other algorithms. For example, it can be executed with a small number of grid points p to provide an initial guess for another heuristic algorithm [57].

3.1.4.2 Random Search

Random search is a direct search method that does not require derivatives to search a continuous domain [58]. The strategy of the random search is to sample solutions from across the entire search space using a uniform probability distribution. Each sample is independent of the previous samples. Random search can return a reasonable approximation of the optimal solution within a reasonable time under low problem dimensionality, although the approach does not scale well with problem size (such as the number of dimensions).

For the ABS placement optimization problem, each sample solution is the tuple of all ABSs coordinates in playground.

3.1.4.3 Nelder-Mead Simplex

Another gradient-free heuristic, suggested in the literature for the placement problems, is the Nelder-Mead simplex algorithm. It is a commonly applied gradient-free numerical method [59]. The Nelder-Mead simplex algorithm uses the concept of a simplex, which is a bounded convex polytope of $p + 1$ vertices in p dimensions. For example, in two-dimensional space, the simplex is a triangle as shown in Fig. 3.6. In each iteration, the algorithm evaluates a function at the vertices of the simplex, then iteratively modifies the simplex based on function evaluation results. The simplex can be modified using shrink, reflection, expansion and contraction operations as shown in Fig. 3.6. The iteration of the algorithm is repeated until some desired bound on the solution quality

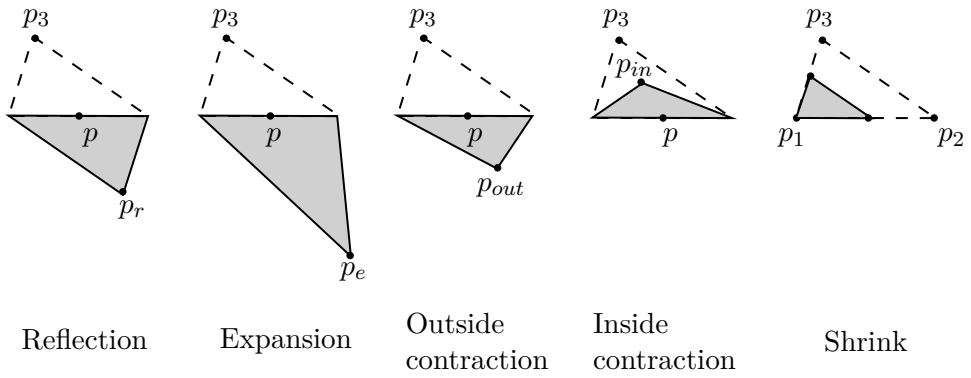


Figure 3.6. – Graphical representation of Nelder-Mead Simplex method operations.

is obtained [60]. Detailed information on the Nelder-Mead simplex algorithm and its realization can be found in [61].

Each vertex of the simplex represents a solution for the optimization problem. For the ABS placement optimization problem, the solution is a tuple of ABSs coordinates in two-dimensional playground. The tuple is used to calculate the aggregated system capacity. A single ABS in two-dimensional space has two degrees of freedom, thus the simplex has three vertices as shown in Fig. 3.6. These three vertices depict the aggregated system capacity for three different solutions.

Multiple ABSs can be located in space independently, increasing the number of degrees of freedom. With each additional ABS, the simplex becomes two additional vertices. Thus, the simplex has $2m + 1$ vertices for m ABSs in two-dimensional playground.

3.1.4.4 Particle Swarm Optimization

In [19] authors propose to use a PSO method in order to find a solution for the ABS placement problem. Inspired by the social behavior of animals, such as fish schooling and bird flocking, PSO is an evolutionary algorithm, which iteratively tries in parallel to improve candidate solutions called particles, which are organized in swarms. The movement of a particle is an aggregated acceleration towards its best previously visited position H^l_{best} and towards the best particle of a topological neighborhood H^g_{best}. In each iteration the local position of particle H^l in the search space is updated by adding the current velocity V^l (please note that the velocity of each particle reflects the distance traveled by this particle at each iteration):

$$H^l(t+1) = H^l(t) + V^l(t+1) \tag{3.6}$$

The velocity includes stochastic and deterministic components. It is updated as follows:

$$V^l(t+1) = \phi V^l(t) + \psi_1 \kappa_1 (H^l_{best}(t) - H^l(t)) + \psi_2 \kappa_2 (H^g_{best}(t) - H^l(t)), \tag{3.7}$$

where H^l_{best} and H^g_{best} denote the best previous positions of the particle and of the swarm, respectively. The weights ψ_1 and ψ_2 are coefficients that determine the acceleration of the particle towards its own and the swarm goal, respectively. The weight ϕ is the inertia weight, which is used to control speed of convergence. The stochastic components κ_1 and κ_2 are uniformly chosen from the interval $[0, 1]$.

Each particle represents a solution for the optimization problem. For the ABS placement optimization problem, the solution is a tuple of ABSs coordinates in two-dimensional playground. The tuple is used to calculate the aggregated system capacity.

The PSO generates multiple particles to search different regions of the solution space independently. Initially, the particles are distributed uniformly over the solution space so that they can sample over most regions. Each particle has multiple degrees of freedom as each ABS can be placed independently of others. The number of degrees of freedom depends on the number of ABSs.

To avoid the local optimum in complex optimization problems, it is necessary to increase the number of particles. However, there are no specific rules to determine the number of particles, as well as their parameters. Also, with the increase in the number of particles, the complexity of each iteration increases.

3.2 Approximated Distance-Based Capacity Optimization

If the solution of a problem is complex and computationally intensive, an approximation can be introduced. An approximated problem is a mathematical problem that mimics the behavior of the initial accurate problem, i.e. the target function, as closely as possible while being computationally cheaper to solve [54].

The SINR-based capacity optimization problem, formulated in Section 3.1, considers the downlink interference, created by ABSs on the same frequency resource. As we have seen, the capacity function is non-concave and non-smooth, which complicates the search for the solution. From the analysis in Section 3.1.3 we have seen that the capacity depends on the distances between the network entities (ABSs and UEs):

- an ABS should be located as close to UEs as possible to maximize the SINR ratio;
- a UEs, served by the particular ABS, should be separated from interfering ABSs to minimize interference.

3.2.1 Problem Approximation

To reduce complexity and assuming that the transmission range of the ABS can be approximated as a convex shape, it is possible to approximate the SINR-based capacity using euclidean distances as shown in Fig. 3.7, where:

- The received power $S_{ij}(d_{ij})$ of UE is approximated by the euclidean distance d_{ij} between the ABS i and UE j;
- The cumulative interference $\sum_{l \in s, l \neq i} I(d_{il})$ for UEs, served by a particular ABS i, is approximated by the distance d_{il} between the serving ABS i and interfering ABSs $\forall l \in [1, m], l \neq i$.

First, as we show in the next section, this approximation reduces the model evaluation complexity. Second, it is possible to formulate the search problem as a planar clustering problem for which fast heuristic algorithms are known [62]. The clustering problem is formulated for our case as follows: a set of UE coordinates u need to be partitioned into m clusters $s = \{s_1, s_2, ..., s_m\}$ so as to minimize the within-cluster variance, i.e. variance:

$$\underset{s}{\operatorname{argmin}} \quad \sum_{j=1}^{m} \sum_{i=1}^{n} \|u_i - s_j\|^2 \tag{3.8}$$

where m is the number of user clusters (which corresponds to the number of ABSs), n is the number of UEs, u_i and s_j - 2D coordinates of the i-th UE and j-th centroid correspondingly. As the formulated clustering problem minimizes variance inside clusters, it also implicitly maximizes the inter-cluster distance.

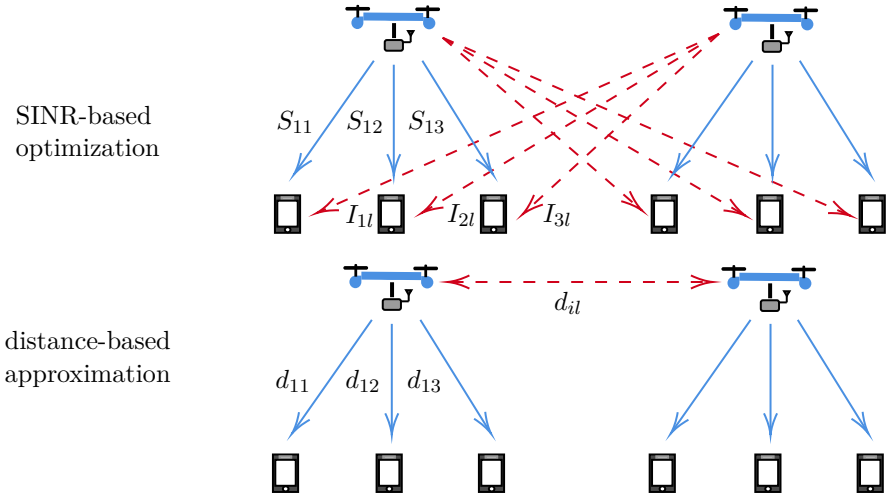


Figure 3.7. – SINR-based capacity and approximation using euclidean distances.

The introduced distance-based approximation decreases the number of local minima in obtained model and reduces the evaluation cost of the target function compared to the accurate capacity evaluation. On the one hand, it allows optimization algorithms to find good solutions more reliably for the approximated problem, reducing algorithm error. On the other hand, the distance-based approximation only implicitly considers the impact of interfering ABSs and signal attenuation with the distance. Thus, the modeling error grows, which can potentially lead to sub-optimal solutions in some scenarios.

3.2.2 Computational Complexity

Finding the exact solution for the clustering problem in Eq. 3.8 is proven to be NP-hard in [63]. As for the complexity of the solution evaluation on the distance-based model, it can be done in polynomial deterministic time assuming the constant cost c_{DIST} of the distance calculation between two points in a plane. As shown in Fig. 3.8, we need to make rm operations to calculate the distances between UEs and their serving ABSs and $m(m-1)$ operations to calculate the distances between ABSs. However, the latter contain repeating operations, and without them only m operations are needed. Thus, the total complexity is $O(rm + m) = O(rm)$.

3.2.3 K-Means Algorithm for the Clustering Problem

The most common heuristics for solving the clustering problem is Lloyd's algorithm, often called K-Means algorithm [62]. It uses an iterative refinement technique and consists of assignment and update steps:

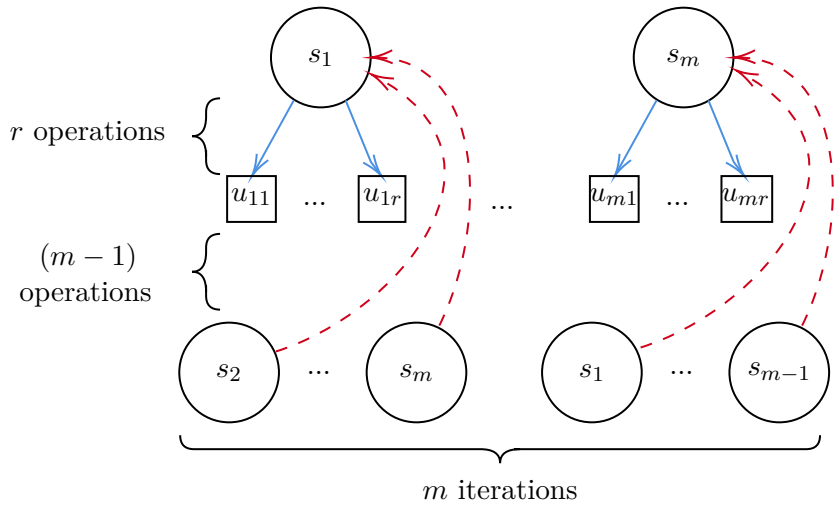


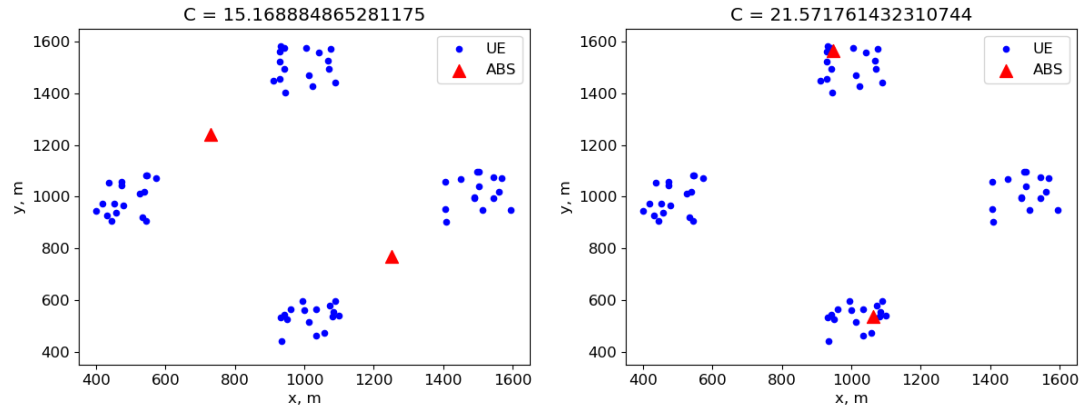
Figure 3.8. – The complexity of a solution evaluation on the distance-based model.

1. Uniformly at random choose initial k centers (centroids) $Z = \{z_1, z_2, ...z_k\}$.
2. Assignment: for each $i \in (1, ..., k)$, assign the cluster to be the set of UEs in u that are closer to z_i than they are to z_j for all $j \neq i$.
3. Update: for each $i \in (1, ..., k)$, set z_i to be the center of mass for all UEs assigned to z_i: $z_i = \frac{1}{|z_i|} \sum_{u \in z_i} u$.
4. Repeat steps 2 and 3 until Z no longer changes.

Due to minimization of the variance in clusters, the algorithm implicitly maximizes the distance between cluster centroids. The complexity of the algorithm is $O(ink)$, where i is the number of algorithm iterations, n is the number of UEs, k is the number of clusters. In practice, the K-Means algorithm quickly converges and thus has an average complexity of $O(nk)$.

3.2.4 Limitations of K-Means Algorithm for ABS Placement

In [26] the authors propose to employ a number of ABSs as the k value. However, if the number of clusters k is not corresponding to the real number of user clusters, then the locations of ABSs are sub-optimal. One example of the sub-optimal K-Means solution for $k = 2$ ABSs, while the real amount of user clusters is 4, is shown in Fig. 3.9(a). Here, UEs from 4 real clusters are assigned into 2 clusters. Because of the relatively high distance between ABS and users, the SINR ratio is degraded which results in reduced capacity.



(a) K-Means solution

(b) Grid-search (near-optimal) solution

Figure 3.9. – Sub-optimal solution of K-Means algorithm in terms of aggregated system capacity C with $k = 2$.

In Fig. 3.9(b) the near-optimal solution obtained by the grid search applied to the SINR-based capacity model is shown. We see that the near-optimal solution in terms of aggregated system capacity is to locate ABSs close to centroids of two user clusters.

3.3 Proposed Hybrid Approach: SINR and Distance-Based Capacity Optimization

To overcome the approximation error of the K-Means algorithm and still provide the fast solution to ABS placement problem, we propose a hybrid heuristics: (1) we first derive a set of feasible solutions iteratively using K-Means algorithm for different k, (2) then choose m out of k center points for each K-Means solution, (3) evaluate these solutions using accurate SINR-based capacity calculation and, finally, (4) we choose the ABS locations, which maximize the aggregated system capacity. The algorithm is shown in Alg. 1.

1: **procedure** PROJECTED CLUSTERING (PC)
2: **Inputs:** UE locations u (size n),
number of ABSs m,
number of available RBs r
3: $k \leftarrow m, C_{best} \leftarrow 0$, $s_{best} \leftarrow \{\}$
/* *Initialization* */
4: **if** PC-Q **then** $k_{max} \leftarrow n - 1$
5: **else if** PC **then** $k_{max} \leftarrow n/r$
6: **end if**
/* *Choose the stopping criterion* */
7: **while** $k \leq k_{max}$ **do**
8: $Z \leftarrow KMeans(u, k)$
/* *K-Means solution:* $O(ink)$ */
9: $s \leftarrow FindFartherstPoints(Z, m)$
/* m *most distant points in* Z*:* $O(klogk)$ */
10: $C \leftarrow FindAggregatedCapacity(s)$
/* *Find the aggregated capacity on the SINR-based model:* $O(rm^2)$ */
11: **if** $C > C_{best}$ **then** $C_{best} \leftarrow C$, $s_{best} \leftarrow s$
12: **end if**
/* *Save current best solution* */
13: $k \leftarrow k + 1$
14: **end while**
15: **return** s_{best}
/* *Return best solution* */
16: **end procedure**

Algorithm 1. – Projected Clustering Algorithm

First, in line 3 we set the number of clusters k equal to the number of ABSs m and in line 7 we set the stopping criterion as $k \leq k_{max}$. Iteratively, we analyze the user distribution to find the amount of user clusters. For this, we try different k values. The K-Means clustering algorithm is executed in line 8. The result consists of the approximated centroids. Because the number of centroids is not necessarily corresponding to the number of m, there is a need to choose only m centroids to be assigned with ABSs. To limit the effect of interference, in line 9 we choose the mutually distant centroids using a simple assignment algorithm, described in the next section. The obtained solution is then evaluated on the accurate SINR-based capacity and the aggregated capacity value for ABSs is calculated (line 10). The best value of the aggregated capacity and corresponding ABSs locations are saved. Finally, the optimized ABS locations, which give the maximum aggregated capacity, are returned as output (line 15).

The presented algorithm minimizes the number of costly evaluations of the accurate SINR-based capacity ($O(rm^2)$ each evaluation) and at the same time avoids sub-optimal solutions derived by the pure clustering algorithm. The example of the near-optimal solution offered by the Projected Clustering (PC) algorithm for the same setting as in the previous section is shown in Fig. 3.10 (compare with Fig. 3.9). In this particular case, the solution differs only minimally from the near-optimal solution of the grid search.

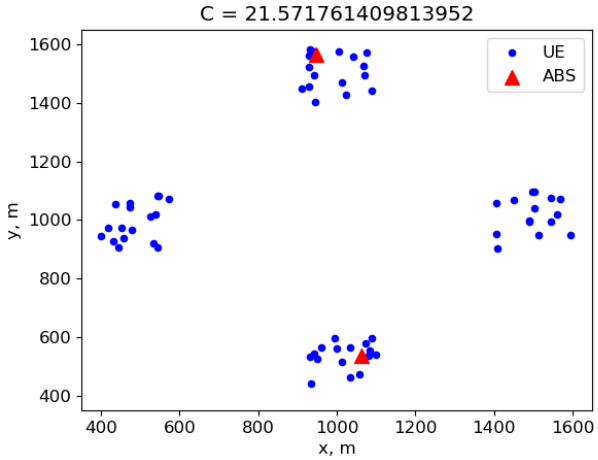


Figure 3.10. – Near-optimal solution of the PC algorithm in terms of aggregated system capacity C.

3.3.1 Assignment of Centroids to ABS Locations

Depending on the input parameter k of the K-Means, the output number of centroids may be greater than the number of ABSs. To choose ABS locations, which minimize the effect of interference, we pick the mutually distant centroids. We use the following algorithm, which is given in Alg. 2:

1. sort the k centroids in Z with respect to y coordinate using the *Heapsort* [64] (complexity of $O(klogk)$);
2. pick the centroid with minimum y coordinate and include it into the solution set;
3. calculate the distances between centroids in the solution set and other remaining centroids which have not yet been calculated;
4. include the centroid with the largest distance sum into the solution set;
5. repeat steps 3 and 4 until m points are chosen.

```
1: procedure FindFartherstPoints
2:     Inputs:  Set of centroids Z (size k),
                number of ABSs m
3:     s ← 0
4:     Z' ← HeapSort(Z, by axis = y)                    ▷ O(klogk)
5:     s ← min(Z', by axis = y)
6:     remove min(Z', by axis = y) from Z'
7:     while len(s) ≤ m do
8:         T ← distance matrix between s and Z'
9:         s ← max(T)
10:        remove max(T) from Z'
11:    end while
12:    return s
13: end procedure
```

Algorithm 2. – Assignment Algorithm

The complexity of this algorithm is $O(km) + O(klogk)$. Since the m is constant in this algorithm, the complexity is $O(klogk)$.

3.3.2 Evaluation of the Stopping Criterion

The number of iterations in the proposed algorithm is defined by the stopping criterion k_{max}. The algorithm evaluates the aggregated capacity in each iteration and selects the best solution. In Fig. 3.11 the best solutions found on each iteration are given

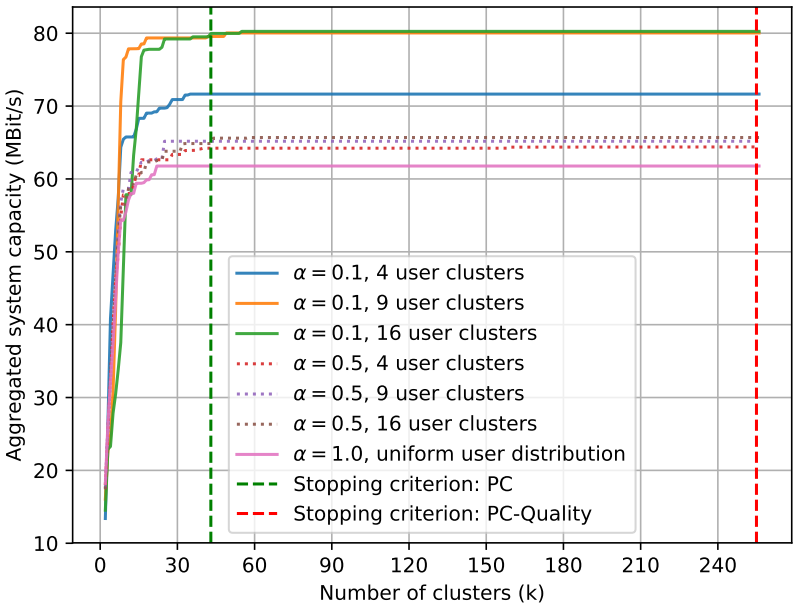


Figure 3.11. – Comparison of two stopping criteria for the PC and Projected Clustering Quality (PC-Quality) algorithms for different user distributions (8 ABSs, 6 RBs, 256 UEs, $\gamma = 2.5$).

for different user distributions. As the number of clusters k increases, the K-Means algorithm can achieve better clustering results. Better clustering results imply better positioning of the ABSs. As shown in Fig. 3.11, the capacity significantly increases until certain k and remains almost constant after. This critical value of k depends on user distribution and increases as user distribution tends to be clustered.

We propose two alternative stopping criteria, which are also shown in Fig. 3.11:

- PC-Quality: $k_{max} = n - 1$. In fact, this stop criterion evaluates all possible user cluster combinations, but needs more time.

- Projected Clustering (PC): $k_{max} = n/r$; the minimum amount of users in the cluster is the number of available RBs for one ABS.

The figure shows that the PC-Quality variant can only find a slightly better solution compared to the PC. For example, the PC-Quality shows 0.35% relative improvement over the PC (80.25 vs. 79.97 Mbit/s respectively) for 16 dense user clusters ($\alpha = 0.1$), while it requires significantly more costly capacity evaluations (255 vs. 42 values respectively).

3.3.3 Complexity of the Projected Clustering

As shown in Alg. 1, the total complexity of the proposed algorithm is as follows:

$$O(ink) + O(klogk) + O(rm^2) \tag{3.9}$$

Taking into account the stopping criterion, the complexity of the PC-Quality algorithm is as follows:

$$O(in^3) + O(n^2logn) + O(rnm^2) \tag{3.10}$$

The stopping criterion for the PC algorithm reduces the complexity by constant factor r.

3.3.4 Algorithm Limitations

It is necessary to keep in mind some design limitations of the proposed algorithm:

1. the proposed algorithm is designed to work when the number of UEs is bigger than the number of ABSs;

2. the adaptation of the algorithm to use with other radio resource schedulers (for example, RR or Proportional Fair (PF) schedulers) may require the adaptation of the stopping criterion; otherwise, the algorithm may generate sub-optimal clustering results.

As the next step, we compare the proposed algorithm to state-of-the-art algorithms in terms of solution quality and computational efficiency.

3.4 Simulation-based Evaluation

In this section, we compare the previously discussed algorithms and examine their performance in the following scenarios:

- different radio conditions (the path loss exponent γ);
- different user distributions:
  - cluster density α;
  - amount of clusters q^2;
- different problem sizes:
  - number of ABSs m;

– number of UEs n.

We also analyze the impact of the resource scheduling strategy on performance of algorithms.

3.4.1 Scenario

The goal is to find a close-to-optimal placement, which maximizes the aggregated system capacity. Six algorithms were compared:

- the Random Search,
- the Nelder-Mead simplex,
- the K-Means clustering,
- the PSO,
- the proposed PC and
- the proposed PC-Quality algorithms.

We have evaluated several possible configurations of algorithms and selected the most effective ones. Algorithm parameters are summarized in Table 3.1.

The stop criterion t_{max} for the PSO and the Random Search algorithms was chosen to produce a runtime similar to the proposed PC algorithm t_{PC}. An increase in runtime would improve the algorithm results, but in the current work we found the improvements to be minor, when compared to the increase in runtime.

The PSO coefficients $\psi_1 = 0.5$ and $\psi_2 = 2.0$ determine the acceleration of the particle towards its own and the swarm goal, respectively. We discovered that if a particle experiences a slight bias in acceleration towards the swarm goal, it yields more accurate results for the placement problem with high amount of a local minima. The weight ϕ, which is used to control rate of convergence is set to 0 in order to not conflict with the stop criterion t_{max}. $swarm_{size}$ parameter controls the size of the particle swarm. This parameter affects significantly the PSO performance. The bigger the number of ABSs, the higher amount of particles is needed to achieve satisfactory solutions. We found that $swarm_{size} = m \cdot 8$ produces a good tradeoff between runtime and solution quality. The initial locations of particles are randomly drawn from the uniform distribution.

The x_{atol}, f_{atol} and max_{iter} parameters of the Nelder-Mead simplex algorithm are used to define the stop criterion. x_{atol} defines the absolute difference between ABSs coordinates of current and previous solutions that is acceptable for convergence. f_{atol} defines the absolute difference in the aggregated capacity between current and previous

solutions that is acceptable for convergence. When the difference between current and previous solutions is lower than the set threshold, the algorithm finishes its execution. We lowered x_{atol} and f_{atol} artificially in order to not conflict with the stop criterion max_{iter}. max_{iter} defines the maximum number of iterations for the Nelder-Mead algorithm. The maximum number of iterations needed by this algorithm is dependent on the number of ABSs. We found that $max_{iter} = m \cdot 600$ produces a good tradeoff between runtime and solution quality.

The number of clusters k for the K-Means algorithm is set to a number of available ABSs m.

User locations are given and fixed during the simulation run. The transmission power of ABSs is constant. The r-best-CQI resource scheduler ($r = 6$) is used, except for Section 3.4.2.7. In Section 3.4.2.7 we compare results for two scheduling strategies: the r-best-CQI and the RR.

3.4.2 Analysis of Aggregated Capacity and Time Complexity

We analyze the influence of different problem parameters on the aggregated system capacity and the CPU time.

3.4.2.1 Impact of Path Loss Exponent on Aggregated Capacity

The results for the average aggregated capacity for the variable path-loss exponent, 4 dense user clusters and 2 ABSs are presented in Fig. 3.12. The variation of the path-loss exponent is changing the radio conditions. Because of the variation in received and interference powers, the signal-to-noise ratio is affected. Thus, the overall system capacity varies as well.

It can be seen that if the path loss exponent is 2.0, the path attenuation is low and users experience higher interference levels from neighboring ABSs. This decreases the

Table 3.1. – Main algorithm parameters (m is the number of ABSs).

Algorithm	**Parameter Values**
Random Search	$t_{max} = t_{PC}$
Nelder-Mead, Simplex	$x_{atol} = 0.0001$ $f_{atol} = 0.0001$ $max_{iter} = m \cdot 600$
K-Means	$k = m$
PSO	$\psi_1 = 0.5$, $\psi_2 = 2.0$ $\phi = 0$, $t_{max} = t_{pc}$ $swarm_{size} = m \cdot 8$ init distribution: '$random\ uniform$'

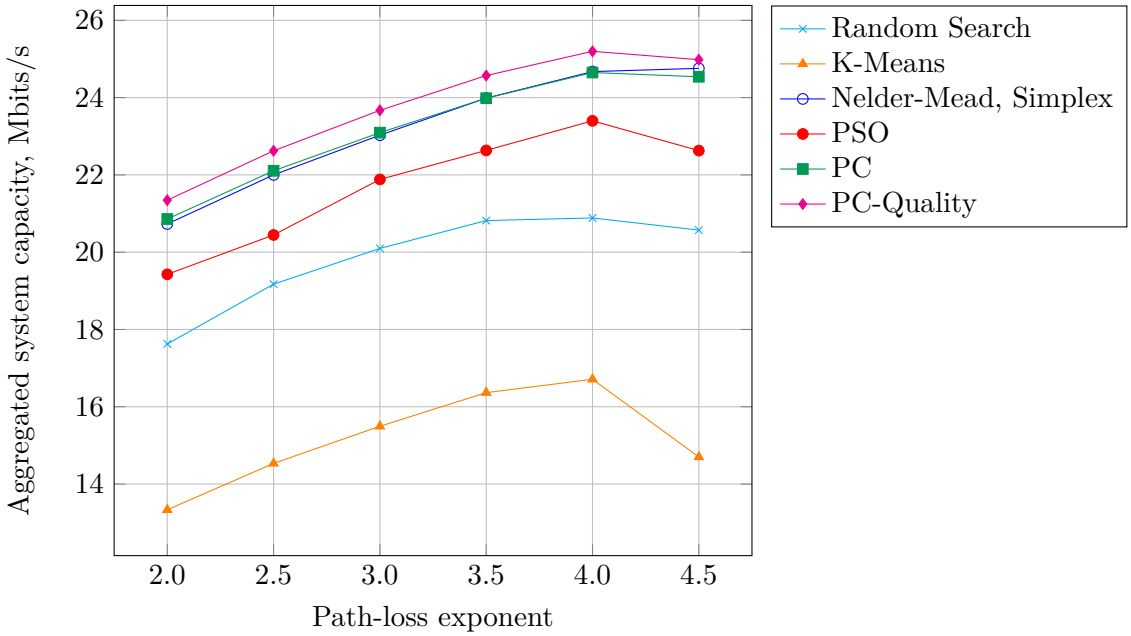


Figure 3.12. – Impact of path loss exponent on average aggregated network capacity for 2 ABSs and 64 UEs, $\alpha = 0.1$, 4 user clusters.

system capacity. With the increase of the path-loss exponent, the path attenuation increases and thus the interference decreases. At $\gamma = 4.5$ we see a drastic decrease in system capacity due to high signal attenuation as a result of higher path losses. The K-Means algorithm shows the greatest degradation due to sub-optimal positioning (ABSs are far away from users).

We can see that in this scenario the K-Means algorithm produces the worst result due to the model approximation error (initialization with the cluster number k of 2).

The Random Search found better solutions as the K-Means ($\sim 32\%$ better on average), but still clearly the sub-optimal. The PSO algorithm shows similar results as the Random Search. The iteration limit and swarm size may be changed to obtain better solutions, but the computational time will increase as well.

The Nelder-Mead algorithm on the accurate SINR-based capacity model gives very good results but requires an enormous amount of CPU time to generate a valid solution.

Our PC algorithm achieves the same result as the Nelder-Mead algorithm. The PC-Quality shows the small increase of the system capacity in comparison to the PC but requires more CPU time.

In general, the change of the path-loss exponent has a minor influence on the individual results of the algorithms ($\sim 10\%$ variance on average). Therefore, we fix the path-loss exponent at $\gamma = 2.5$ for next simulations.

3.4.2.2 Impact of User Distribution on Aggregated Capacity

In order to analyze the impact of the user density on the algorithms performance, in this scenario we create 4 dense user clusters and vary their density α as described in Section 2.2.4.1. The results for the average aggregated capacity for various densities and 8 ABSs are presented in Fig. 3.13. We can see that the K-Means algorithm generates very good solutions, because it can correctly characterize the user distribution (in this case $k = 8$, which is greater than the number of real clusters). The Random Search and the PSO show similar sub-optimal behavior and increase their solution quality as the user distribution tends to be uniform. Good results for uniform distributions ($\alpha = 1.0$) can be explained as follows: when the users are located uniformly, there is a higher chance to find a good solution stochastically.

As before, the PC, the PC-Quality and the Nelder-Mead algorithms show similar performance with small degradation at uniform distributions.

In general, it can be seen that the user density strongly influences the solution quality of the algorithms and that in the cases where the real number of user clusters is less than the number of ABSs, the K-Means show a very decent solutions. For uniform distributions, the algorithms show mostly similar results (around $\sim$ 13% difference

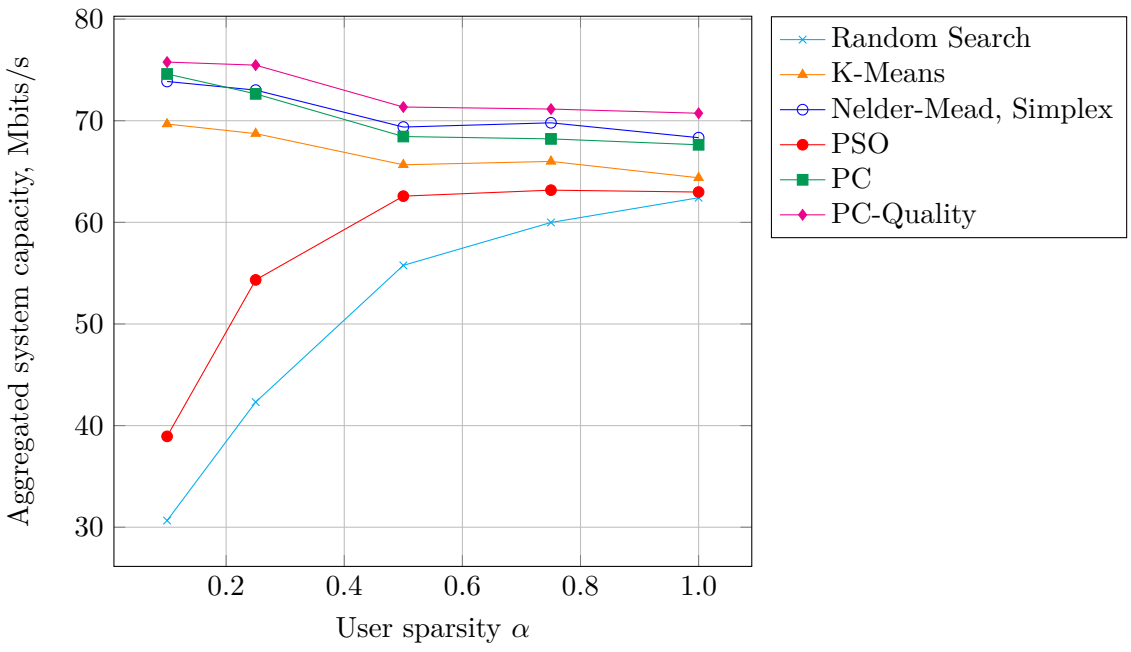


Figure 3.13. – Impact of user density on average aggregated network capacity for 8 ABSs and 256 UEs, $\gamma = 2.5$, 4 user clusters.

between the worst and the best), because there are more chances to find a good subset of UEs.

Thus, we fix the user density at $\alpha = 0.1$ to analyze the impact of changes in the number of user clusters on the performance of algorithms.

3.4.2.3 Impact of Number of User Clusters on Aggregated Capacity and CPU Time

In this scenario we fix the user density at $\alpha = 0.1$. The results for the average aggregated capacity vs. the CPU time for variable number of user clusters and 8 ABSs are presented in Fig. 3.14 and the corresponding Pareto front is shown in Fig. 3.15. The values in these figures depict the number of user clusters.

As we can see from the figure, the PSO and the Random Search provide consistent sub-optimal results for any amount of user clusters, while having similar execution times as the PC algorithm.

The K-Means shows the smallest computational time and gives good solutions for 4 and 9 user clusters ($\sim 100x$ faster than the PC). However, for the 16 user clusters it gives the worst solution from all compared algorithms. The k is set to 8 in this case

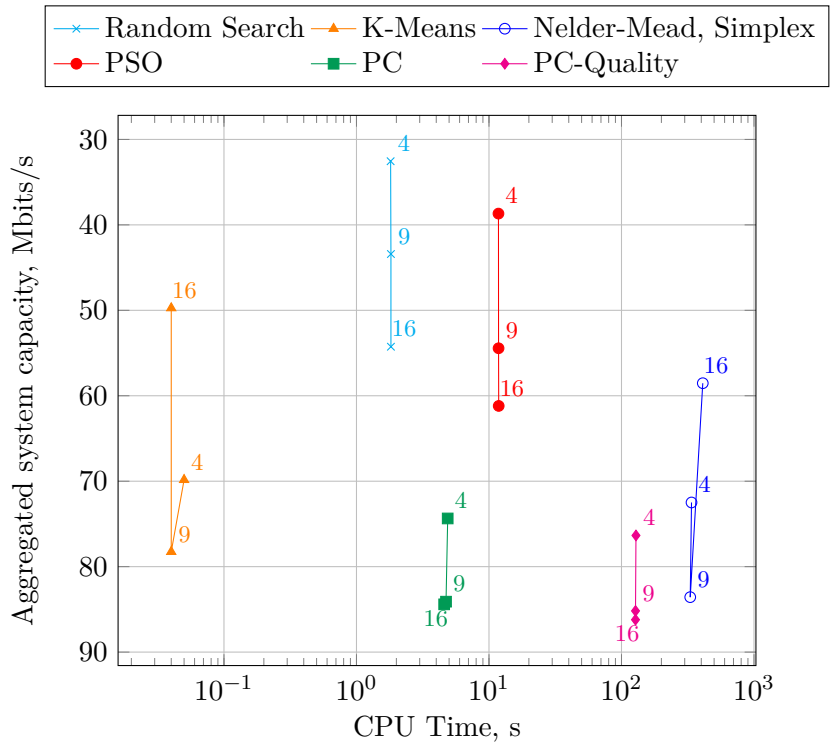


Figure 3.14. – Impact of number of user clusters on average aggregated network capacity and CPU time for 8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$.

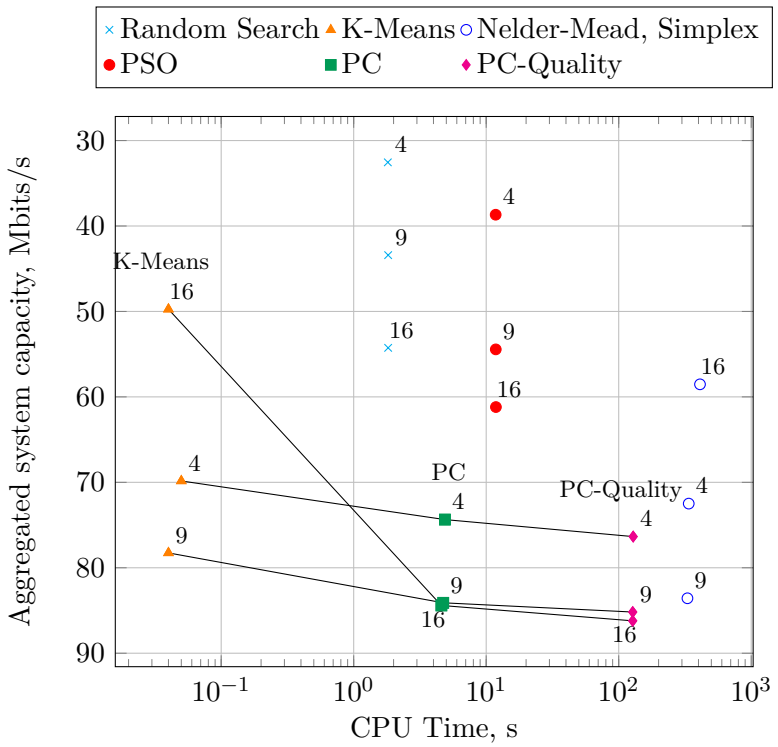


Figure 3.15. – Pareto front: impact of number of user clusters on average aggregated network capacity and CPU time (8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$).

and the K-Means tries to merge several clusters into one. It can be concluded that the K-Means is not the best choice due to the approximation error for the user distributions, where k is less than the real cluster number.

In general, the Nelder-Mead generates good solutions, except for the cluster amount of 16. Its performance degrades due to the high amount of local minimums and the fact that it was not designed for highly non-convex problems. In general, the computational time is $\sim 75x$ bigger than of the PC algorithm.

Our proposed algorithms PC and PC-Quality show consistently good results and perform well also for 16 user clusters. As previously, the PC-Quality provides only minor improvement in the capacity ($\sim 2\%$) for the price of significantly larger computational time ($\sim 26x$). The PC algorithm shows a trade-off between different algorithms: it provides the nearly optimal solutions with relatively small computational times.

The summary of results and their relative comparison is given in Tables 3.2 and 3.3.

Table 3.2. – Absolute and relative comparison of aggregated capacity between algorithms for different amount of user clusters. The results of the PC algorithm are used as a baseline for the relative comparison.

	Aggregated Capacity, $Mbits/s$			**Relative Improvement**, %		
# clusters / **Algorithm**	4	9	16	4	9	16
Random Search	32.55	43.41	54.26	−56.23%	−48.38%	−35.72%
K-Means	69.84	78.26	49.74	−6.08%	−6.94%	−41.07%
Nelder-Mead, Simplex	72.49	83.57	58.53	−2.51%	−0.63%	−30.66%
PSO	38.68	54.44	61.19	−47.98%	−35.27%	−27.51%
PC	74.36	84.10	84.41	0.0%	0.0%	0.0%
PC-Quality	76.35	85.18	86.21	2.68%	1.28%	2.13%

Table 3.3. – Absolute and relative comparison of CPU time between algorithms for different amount of user clusters. The results of the PC algorithm are used as a baseline for the relative comparison.

	CPU Time, s			**Relative Improvement**, times		
# clusters / **Algorithm**	4	9	16	4	9	16
Random Search	1.81	1.82	1.82	0.37	0.38	0.4
K-Means	0.05	0.04	0.04	0.01	0.01	0.01
Nelder-Mead, Simplex	337.67	330.23	410.67	69.19	69.67	89.47
PSO	11.79	11.81	11.84	2.42	2.49	2.58
PC	4.88	4.74	4.59	1	1	1
PC-Quality	128.6	127.58	127.56	26.35	26.92	27.79

3.4.2.4 Impact of Number of Aerial Base Stations on Aggregated Capacity and CPU Time

In order to see the impact of the number of ABSs on algorithms' performance, in this scenario we fix the cluster number at 4. The results for the average aggregated capacity vs. the CPU time for variable number of ABSs and 256 UEs are presented in Fig. 3.16 and the corresponding Pareto front is shown in Fig. 3.17. Values on these figures depict the number of ABSs to be placed.

It can be seen that the K-Means algorithm and our proposed algorithms PC and PC-Quality differ by an almost constant factor in terms of the CPU time, which means that the scalability of the two algorithms is roughly comparable. Nelder-Mead has the biggest computational time, while the Random Search and the PSO algorithms show

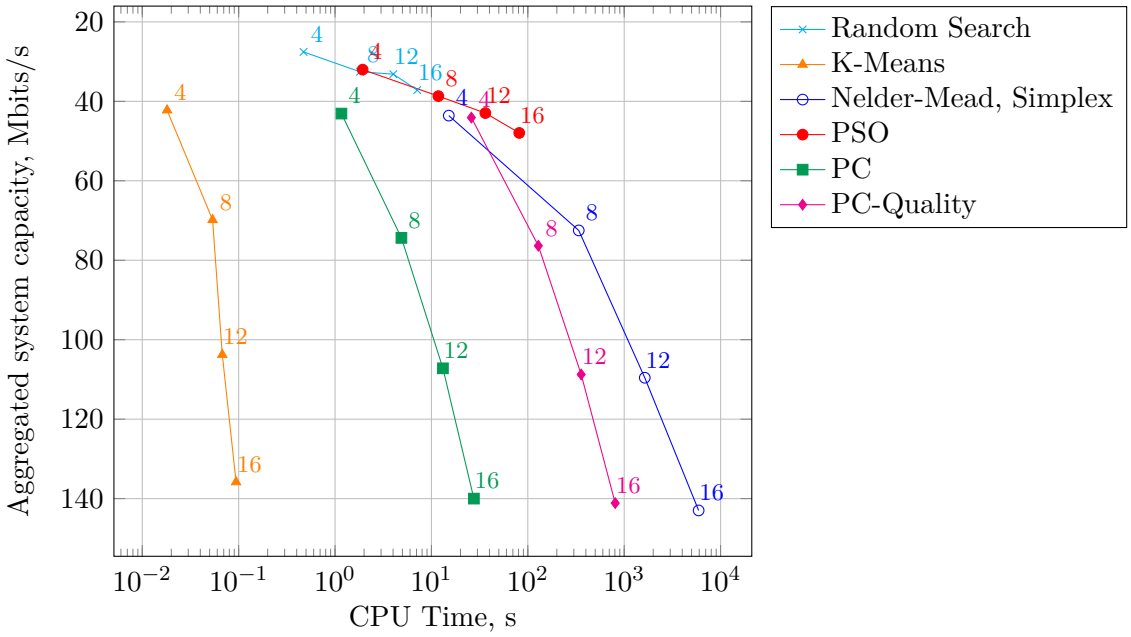


Figure 3.16. – Impact of number of ABSs on average aggregated network capacity and CPU time for 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$, 4 clusters.

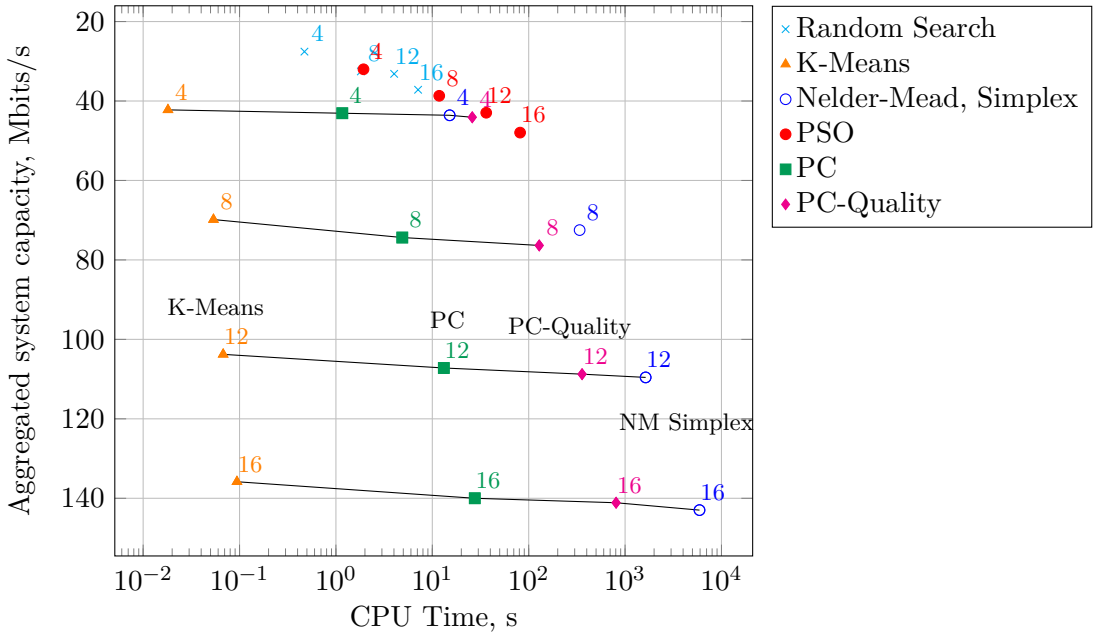


Figure 3.17. – Pareto front: impact of number of ABSs on average aggregated network capacity and CPU time (256 UEs, $\gamma = 2.5$, $\alpha = 0.1$, 4 clusters).

the sub-optimal solutions. An increase in allowed computational time for the Random Search and the PSO would improve their results in terms of aggregated system capacity, but the improvements are minor, when compared to the increase in computational time.

Please note that for 16 ABSs most of the algorithms show high absolute computational time values (except for the pure K-Means algorithms). The proposed PC algorithm needs around 50 seconds and the Nelder-Mead – around 2 hours, which may become a problem in real-time scenarios.

3.4.2.5 Impact of Number of Users on Aggregated Capacity and CPU Time

The results for the average aggregated capacity vs. the CPU time for 8 ABSs and the variable number of UEs are presented in Fig. 3.18 and the corresponding Pareto front is shown in Fig. 3.19. Values on these figures depict the amount of UEs. To simplify these figures, the values for 96, 128, 160, 192 and 224 are omitted.

As before, we see that our proposed algorithm PC shows the trade-off between the K-Means and the Nelder-Mead algorithms, achieving stable near-optimal results in a relatively short time. The Random Search and the PSO show the sub-optimal performance.

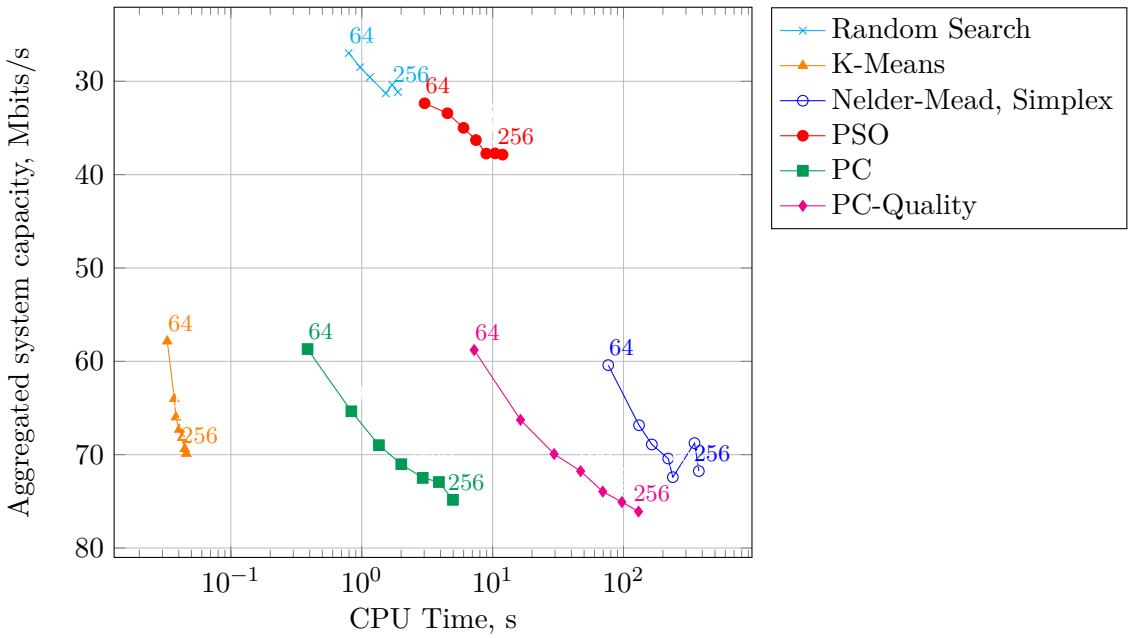


Figure 3.18. – Impact of number of UEs on average aggregated network capacity and CPU time for 8 ABSs, $\gamma = 2.5$, $\alpha = 0.1$, 4 clusters.

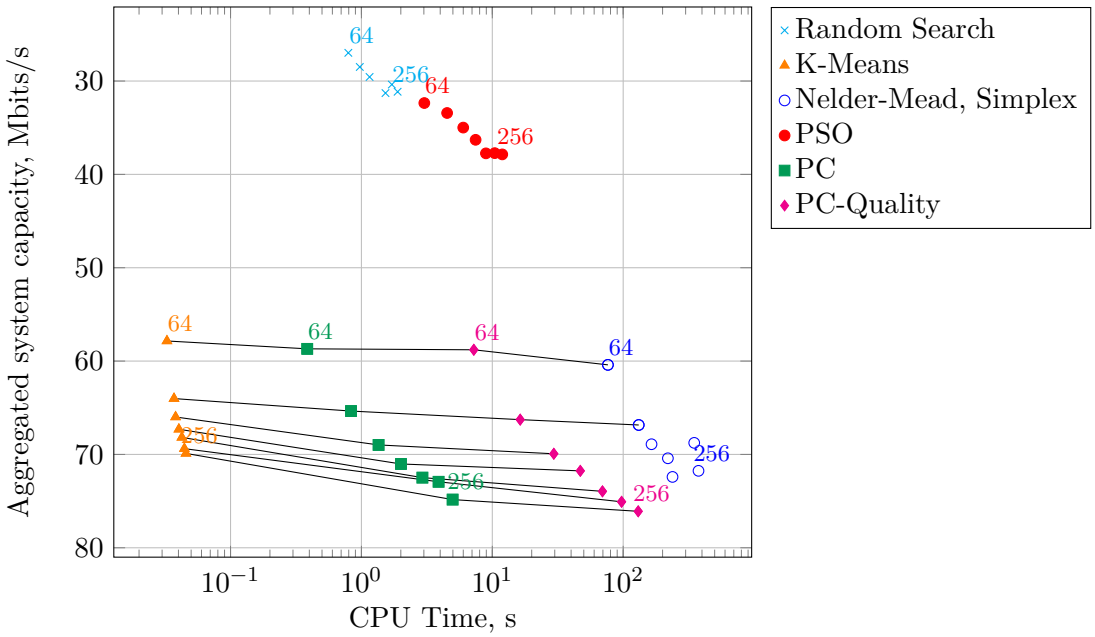


Figure 3.19. – Pareto front: impact of number of UEs on average aggregated network capacity and CPU time (8 ABSs, $\gamma = 2.5$, $\alpha = 0.1$, 4 clusters).

Since the algorithms become more choices to select the best UEs opportunistically and the number of ABS remains the same, we can see a saturation when the number of UEs increases.

It should be noted that the increase in number of UEs also increases the computational time of the proposed PC and PC-Quality algorithms. This result is expected as with the increase in number of UEs, the number of candidate solutions obtained by clustering also increases.

3.4.2.6 Variation in Solution Quality

Previously, we have shown the average values over many simulation runs. In Fig. 3.20 we show the variations in the aggregated system capacity over 64 simulation runs (8 ABSs, 256 UEs and 4 dense user clusters) to estimate stability of the algorithms.

It can be seen that the Random Search and the PSO show relatively large deviations in the quality of the obtained solutions due to their stochastic nature. The PSO shows a high asymmetry in solution quality (lower quartile) and has a significant number of solutions comparable with results of the Random Search. The Nelder-Mead algorithm is able to find near-optimal solutions, however, has outliers and the variance higher then

the PC, the PC-Quality and the K-means algorithms have. The K-means, the PC and the PC-Quality algorithms show the most consistent performance with small variations.

3.4.2.7 Impact of Resource Scheduling Strategy on Aggregated Capacity

In order to analyze the impact of radio resource scheduling strategy on aggregated capacity, in this scenario we fix the number of ABSs at 8, the number of UEs at 256 and the number of UE clusters at 9. The results of the average aggregated capacity for various user densities and two different radio resource scheduling strategies (Round-Robin (RR) and r-best-CQI (BCQI)) are presented in Fig. 3.21.

It is evident that the K-Means, the Nelder-Mead, the PC and the PC-Quality show almost the same results for both scheduler strategies, when the user density is very close to 0. When the user density is 0, all UEs inside their cluster have the same position. In this case, the radio conditions for all UEs inside their cluster are the same. The fair RR and the non-fair r-best-CQI scheduling strategies show similar results in terms of the aggregated capacity. The slight difference between them is resulting from the scenario: 8 ABSs are placed in 9 user clusters. The Random Search and the PSO algorithms are

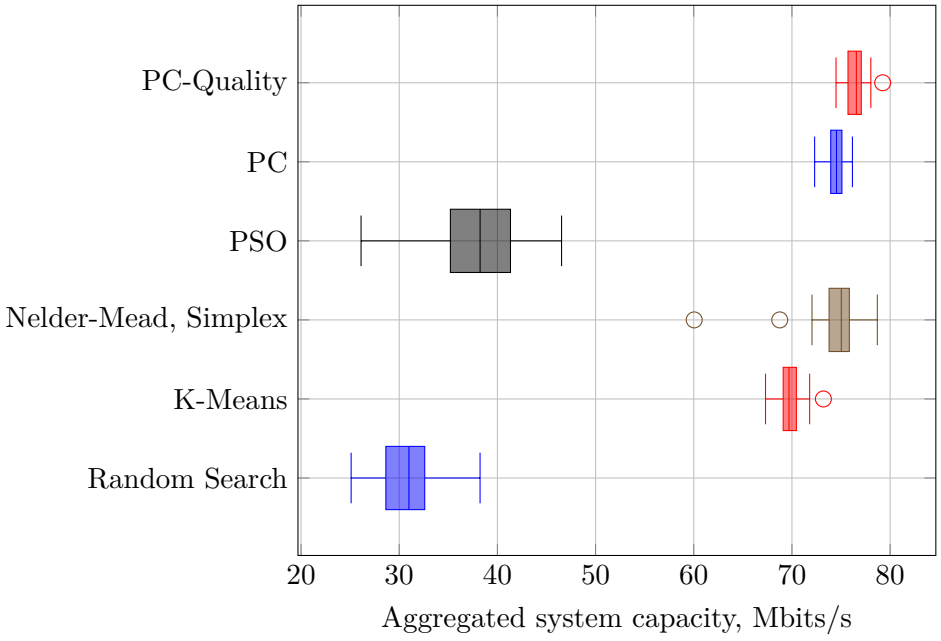


Figure 3.20. – Aggregated network capacity for 8 ABSs, 256 UEs and 4 user clusters, $\gamma = 2.5$, $\alpha = 0.1$. The boxplot displays the distribution of data based on a five values summary: minimum, first quartile (Q1, 25th percentile), median (50th percentile), third quartile (Q3, 75th percentile), and maximum. Outliers are shown as circles.

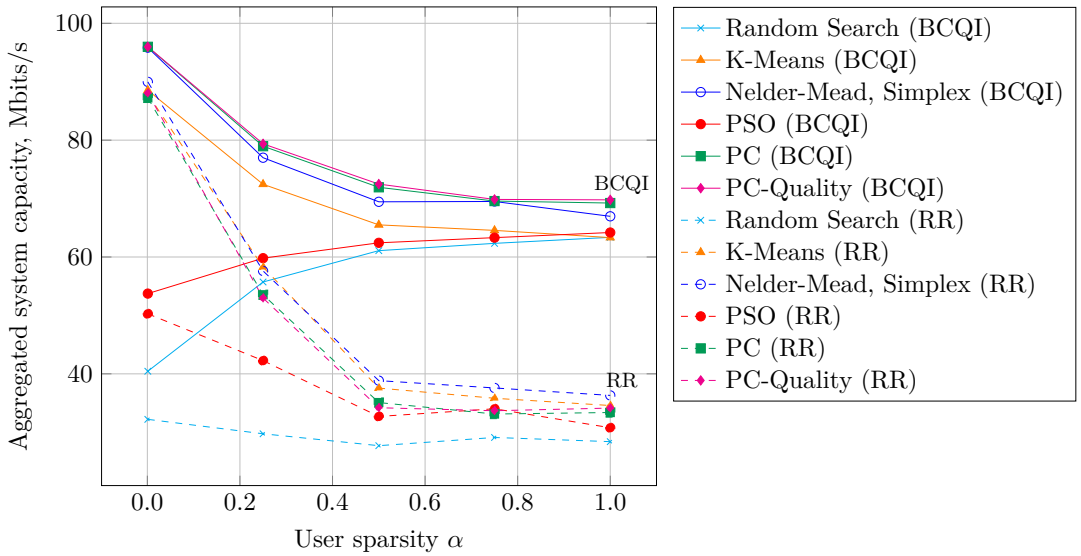


Figure 3.21. – Impact of resource scheduling strategy and user density on average aggregated network capacity for 8 ABSs and 256 UEs, $\gamma = 2.5$, 9 user clusters. Applied schedulers are shown in brackets: Round-Robin (RR) and r-best-CQI (BCQI).

the notable exceptions. Their results are sub-optimal and thus, the aggregated capacity values are not the same for the RR and the r-best-CQI scheduling strategies.

When the user distribution becomes sparser, the aggregated capacity degrades and, finally, converges to two different, but almost constant levels for both RR and r-best-CQI scheduling strategies. The RR scheduler distributes radio resources fairly for all UEs. UEs located at edges between neighboring ABSs, in contrast to UEs located close to the serving ABS, experience higher interference levels and contribute to the lower aggregated capacity.

In contrast, the r-best-CQI scheduling strategy favors UEs with the best channel conditions at expense of worse fairness. Hence, it achieves a higher level of aggregated capacity in comparison to the RR scheduling strategy.

The proposed PC and PC-Quality algorithms derive slightly worse solutions in terms of aggregated capacity, when the RR scheduling strategy is used. The goal of the proposed PC and PC-Quality is to maximize the offloading capacity and they find ABSs locations, which are better suited for the r-best-CQI scheduler. Therefore, the near-optimal solution for the r-best-CQI scheduler may be sub-optimal for the RR scheduler.

3.5 Conclusion

In this chapter, we have shown that the aggregated system capacity can be significantly increased with the optimized placement of aerial base stations. We presented a problem formulation for the ABS placement optimization problem in traffic offloading scenario and showed that this problem is intractable and requires heuristic optimization algorithms. Model properties, which can be used for the problem regularization and relaxation, have been presented as well.

We have proposed a novel hybrid algorithm, which quickly derives multiple solutions on the approximated distance-based capacity model and evaluates them on the accurate SINR-based capacity model, validating the solution. Simulation results showed that, in contrast to the state-of-the-art methods, our approach achieves near-optimal solutions by introducing only a minor increase in computational complexity in comparison to the fastest K-Means algorithm. This ensures that our algorithm can deliver fast and accurate ABS position updates in dynamic scenarios.

Our studies leads to the following main conclusions:

1. the aggregated system capacity can be significantly improved by optimized base station placement;
2. the formulated ABS placement optimization problem is complex and requires exponential time to find the exact solution;
3. the optimization problem in 2D space can be approximated by the distance-based capacity model, which can be solved computationally efficient by clustering algorithms;
4. the K-Means algorithm results in a significant error in scenarios where the number of ABSs is smaller than the number of user clusters;
5. the path-loss exponent affects the individual performance of optimization algorithms only to a small extent ($\sim 10\%$ variance);
6. user distribution plays a significant role in the optimization
   - the clustered user distribution results in many local minimums, i.e. the sub-optimal solutions of state-of-the-art algorithms ($\sim 52\%$ variance in the aggregated system capacity);
   - the uniform user distribution results in similar solutions as state-of-the-art algorithms ($\sim 10\%$ variance);
7. the proposed hybrid algorithm PC:

- derives the candidate solutions on the distance-based capacity model using K-Means clustering algorithm and chooses the best solution based on the accurate SINR-based capacity model;
- is $\sim 89x$ faster and achieves $\sim 31\%$ higher system capacity than the Nelder-Mead algorithm;
- is $\sim 100x$ slower, but achieves $\sim 41\%$ higher system capacity than the K-Means algorithm.

So far, we have discussed only centralized ABS placement optimization, where the central decision-maker is assumed to have global knowledge. In the next chapter, we discuss cases where this requirement does not hold and how we can approach the problem given only local knowledge.

Chapter 4

Self-Organized Aerial Base Station Placement

> The central enemy of reliability is complexity.
>
> Geer et al.

In the previous chapter we discussed centralized algorithms to solve the stationary ABS placement problem. In a real scenario, there is an additional problem of data transmission and synchronization of ABSs. In this chapter we discuss these problems and propose the solution using a self-organizing framework that can increase the overall reliability and scalability of the system.

The proposed SO framework called the Autonomous Infrastructure Deployment Algorithm (AIDA) is designed to solve the ABS placement problem and maximize the aggregated system capacity without relying on long-range transmissions, using only locally available observations.

4.1 Dynamic ABS Placement Problem

In the previous chapter we focused on the stationary ABS placement optimization problem, assuming that the global information about UEs' locations is available. In a real operation, this information needs to be collected before solving the optimization problem as shown in Fig. 4.1. Moreover, after the optimization process, the solution, which contains target positions of ABSs, needs to be delivered to them. In general, three operations can be distinguished:

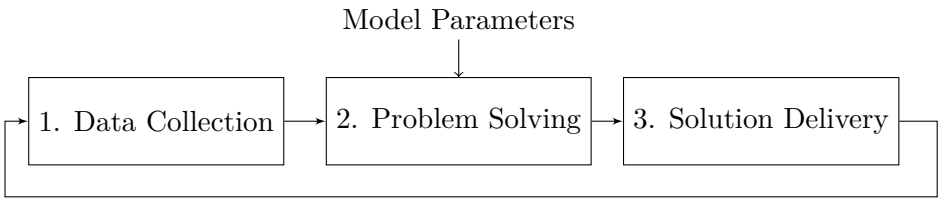


Figure 4.1. – A continuous process of solving the ABS placement problem.

1. Data Collection: the optimization task should have a set of valid input data, namely, the UEs' positions. This step includes the transfer of this data from UEs to the ABSs and then to the central decision-maker and it introduces a data transmission delay as well as network overhead. Network overload and unreliable wireless medium may contribute to the increase in the data transmission delay and may increase the probability of the outdated data to be transferred.

2. Problem Solving: at this step, the decision-maker centrally solves a large-scale problem on the dedicated processing server. As we have seen from the previous chapter, centralized algorithms already require a significant time for small problem sizes (for the network with 256 UEs and 16 ABSs the proposed PC needs around 50 seconds). With the increase of the number of UEs and ABSs, the computational time increases, as it is shown in Fig. 3.16 and 3.18. Thus, centralized algorithms have limited scalability. Another problem is the reliability of this system: in the event of failure of the centralized decision-maker or its communication link, the entire system fails. Also, if one of the ABSs fails or leaves for recharging, the solution of the centralized algorithm must be recalculated.

3. Solution Delivery: the solution of the optimization problem – the target coordinates are sent to the corresponding ABSs. As in the first step, the data transmission delay and network overhead are introduced. Besides, the transmission link between the particular ABS and the centralized decision-maker may fail.

Furthermore, UEs may move in the area, and thus a mission should be divided into small enough time intervals to adapt the solution. This inevitably involves high computational and communication demands on the system.

One way to reduce these demands is to distribute the optimization task between ABSs instead of using a centralized decision-maker. In Fig. 4.2 the classification of systems by their degree of centralization is given with examples for each type. In distributed systems with explicit coordination, the supervisor process distributes tasks to computing nodes and collects the results. This architecture is very common for grid computing, where the large task can be parallelized and accelerated using many separate computing units.

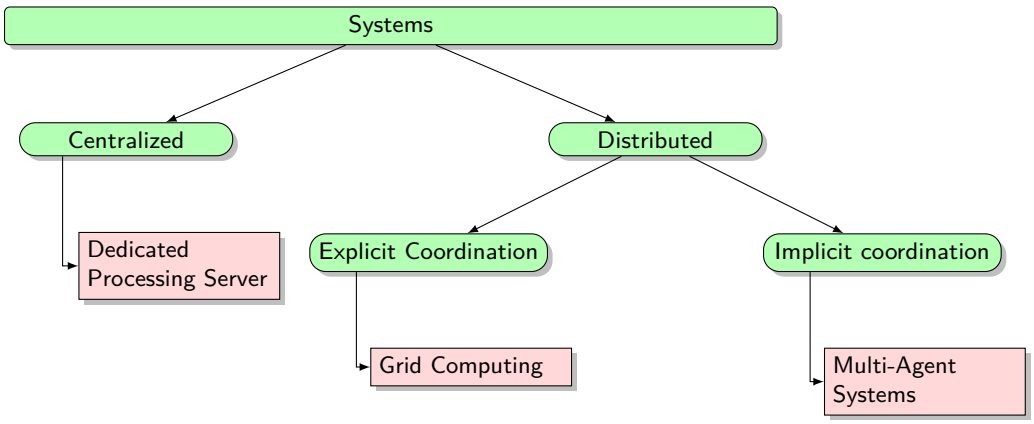


Figure 4.2. – Classification of system types by degree of centralization and their examples.

This approach can distribute the computational load, but has no graceful degradation (if the supervisor process fails, then the whole system fails) and generates network overhead, as well as introduces communication latency.

Regarding the control of ABSs, distributed systems with implicit coordination are of bigger interest. In these systems, each ABS can determine its motion according to its local state, its local perception of environment and its interactions with other ABSs. First, this approach distributes the computational load over multiple ABSs and reduces the communication needs. As a result, distributed control of the ABS system is highly scalable to large systems and adaptive to unknown and dynamic environments and changes in the system itself. With properly designed distributed control laws, the desired global goal of an ABS system can be achieved as the combined outcome of self-deployment of individual ABSs. Second, the system is more resistant to failure: in the event of a failure of one of the ABS, other ABSs can detect it through the environment and take over the whole or a portion of its functions.

Such systems are described as Multi-Agent Systems (MASs) or Self-Organized (SO) systems. They consist of multiple interacting intelligent agents, each solving a particular optimization task, and together they search for the solutions to the problems that are beyond the individual capabilities of knowledge of each agent [65].

The agents in MASs have several important characteristics [66]:

- autonomy: agents are self-aware, operate autonomously without the direct intervention of other entities and perform actions according to their internal state;
- reactivity: agents are part of the environment and can perceive it and respond to changes that occur in it;
- proactiveness: the reaction of the agent should be goal-oriented;

- social ability: agents interact with other agents;
- local views: no agent has a full global view on the system.

To successfully apply a MAS systems, it is necessary to identify agents and their environment, define interactions between them and control rules for agents.

4.2 Proposed Self-Organized Placement Framework AIDA

In the proposed self-organized ABS placement framework, called AIDA, we define the following entities:

- Agents: In our case, each ABS is a self-contained substance or an *intelligent agent*. Each ABS collects location information about ABSs and UEs in the vicinity (defined by its transmission/reception range), conducts decision-making processes and selects actions (changing its position).
- Environment: agents interact continually with their environment, which consists of other agents and UEs. The environment is responding to actions of the agent and presenting new situations to the agent. The agent can gain limited information about the environment or, in other words, has own perception of environment.

In the proposed framework, each agent (ABS) has *egoistic* behavior, e.g. tries to maximize its own capacity. The individual capacity, which can be achieved by an ABS depends on the proximity to UEs and interference from neighboring ABSs. Thus, each selfish agent should seek to minimize distance to users and maximize distance to neighboring agents in order to increase its own capacity. We expect these actions to be preferred by the vast majority of agents, which will lead to their commitment in the increase of the overall system capacity.

4.2.1 Agent Control Rules

Each agent receives information from the environment using sensors (in our case the radio subsystem) and then estimates its actual state. At each time step, the agent implements a mapping from state representation to probabilities of selecting each possible action. This mapping is called the agent's control rules. Finally, the motion control gives the actuators a signal to implement the selected action.

In order to conduct decision-making, control rules for agents need to be defined. These rules are based on the agent's knowledge about the external environment, which is limited locally for each agent. It is possible to distinguish two agents classes: reactive and deliberative agents. Reactive agents simply execute the predefined behavior (similar

to reflexes in biological organisms) without regard to an internal state. Thus, reactive agents orient themselves to the achievement of short-term goals and use simple behavior rules. Literature provides many examples of control rules for reactive agents, inspired from physical processes: potential/force field methods [67,68], kinetic theory of gases [69], diffusion-based [70] or fluid dynamics-based [71] methods. The agents convert the data obtained from sensors (for example, positions of other agents and obstacles) to motion vectors for avoiding obstacles, avoiding other agents, moving to a goal location and maintaining the desired formation. The agent movement is a simple weighted sum of these motion vectors. The orientation on short-term goals increases the probability of reactive agents being trapped in local minima inside non-convex environments.

On the other hand, deliberative agents try to maintain their internal state, predict the effect of possible actions to find optimal way to reach the goal. Deliberative agents are oriented to the achievement of long-term goals and can successfully operate in non-convex environments. However, deliberative agents need to have an accurate model of the environment and may require more computational time [72, 73]. In partially observable environments, the learning of the state-action space is used to ensure the correctness of the model of environment [74, 75]. However, such approaches are trained for specific scenarios only, and re-training requires significant time.

We use the hybrid agent, which includes deliberative and reactive behavior. The agent design is shown in Fig. 4.3. Deliberative behaviour is used to learn and identify the situation, which the agent is experiencing by using its internal perception of the environment. To reduce the state space for learning, we use deliberative behavior only to learn when to switch between three main types of control rules. These control rules

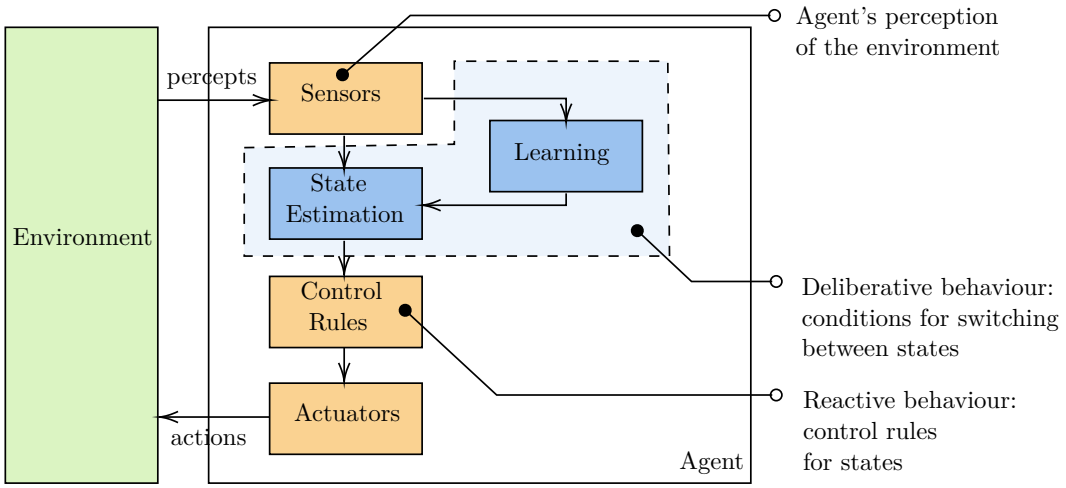


Figure 4.3. – AIDA: agent structure.

are given for three main situations, which define the internal state of the agent as shown in Fig. 4.4:

1. Exploration state – no location information about UE or other ABSs is available. In this state, the ABS explores the environment. As the ABS senses the presence of neighboring ABSs (local set of ABSs s_l is not empty), it proceeds with transition $T0$ into the *separation* state. As UEs get connected to the ABS (local set of users u_l is not empty), it proceeds with transition $T1$ into the *placement* state.

2. Separation state – conflict situation when several ABSs work for the same area at the same time. In this state, care is taken that ABSs do not create strong downlink interference. As the interference measure is below the threshold D, the ABS proceeds with transition $T2$ into the *placement* state (note: there are should be UEs in vicinity).

3. Placement state – ABSs are well separated in the space. Here, one of previously discussed centralized algorithm is applied to the sub-problem, finding the optimal position for the ABS. As the interference measure is above the threshold D, the ABS proceeds with transition $T3$ into the *separation* state (note: there are should be neighboring ABSs)

Each state consists of specific control rules and implements the reactive agent behaviour. Next, we define the agent reactive behavior in each state and describe state transitions in detail.

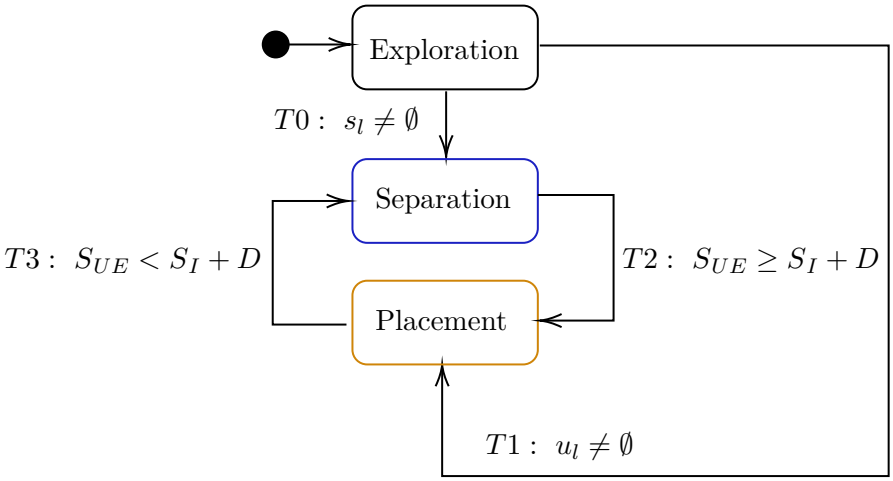


Figure 4.4. – The simplified automata for the decision-making in AIDA framework.

4.2.1.1 Exploration State

In the exploration state, when no information about other nodes in the vicinity is available, the area needs to be explored. As shown in [76], the exploration can be successfully done using a random walk. The random walk is a simple and memoryless technique. The ABS randomly chooses a destination waypoint, uniformly distributed and enclosed by the area constraints, and moves towards the chosen waypoint for a predefined time. After arrival, it chooses another random destination.

In real temporary event scenarios, it may be possible to know a rough UE location estimation, which can improve the performance of the initial search in comparison to the random walk. Therefore, the control rules in exploration state for real systems may be replaced with a set of predefined waypoints. This state, however, is used only as a backup in case if the ABS is lost or UEs are located at a very low density.

As the agent senses the presence of neighboring ABSs, it proceeds with transition $T0$ into the *separation* state:

$$T0 : s_l \neq \emptyset, \tag{4.1}$$

where s_l is the local set of neighboring ABSs, As UEs get connected to the ABS , it proceeds with transition $T1$ into the *placement* state.

$$T1 : u_l \neq \emptyset, \tag{4.2}$$

where u_l is the local set of connected UEs.

4.2.1.2 Separation State

The goal of this state is to separate multiple ABSs in the space to limit the downlink interference at UEs.

To achieve the separation in space between ABSs, we propose to use the artificial force field concept. The idea is that charges with the same sign are repulsive and charges with different signs are attractive to each other. This means that each ABS is capable of generating a repulsion vector itself and selecting a direction of movement to achieve a certain separation distance between different ABSs. Therefore, such a system does not require central control and does not lead to unnecessary complexities.

If the ABS fails, then the repulsion force disappears, and other ABSs will no longer take it into account. This means that the gap created by the failure of the ABS will be covered by other ABSs in the vicinity.

Mathematically, the repulsive potential is defined as follows:

$$U(x,y) = \sum_{j=1}^{m} U_{ABS_j}(x,y), \tag{4.3}$$

where U_{ABS_j} is the repulsive potential to neighboring ABS j and m is the number of ABSs in the vicinity. The repulsive potential U_{ABS_j} is as follows:

$$U_{ABS_j}(x,y) = \frac{1}{d^{\mu}}, \tag{4.4}$$

where d is the distance to the j neighboring ABSs in meters and μ is the scaling coefficient, which defines the attenuation of repulsive potential. The example of repulsive potential, calculated for three neigboring ABSs is shown in Fig. 4.5. The repulsive potential is stronger when the ABS is closer to the neigboring ABSs and is decreasing when the ABS is far away.

The repulsive force constrains the ABS to repel from neighboring ABSs and is defined as the negative gradient of the potential function:

$$F(x,y) = -\nabla U(x,y), \tag{4.5}$$

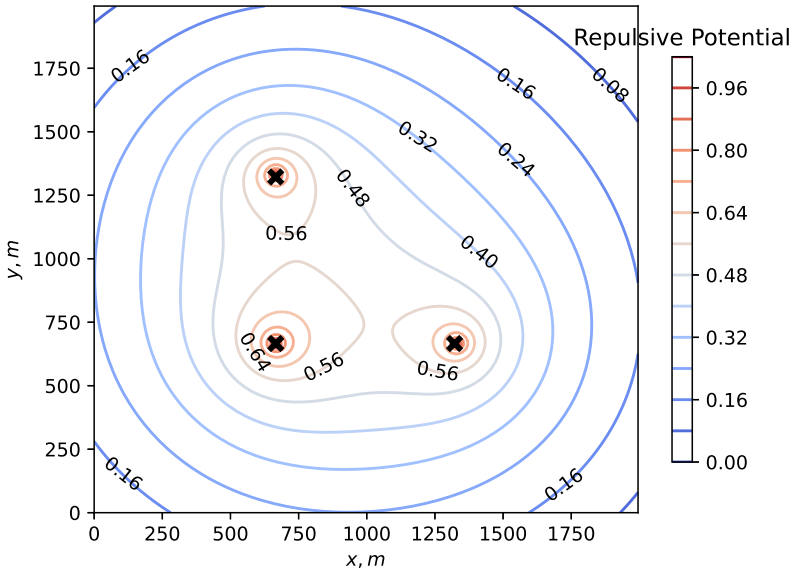


Figure 4.5. – Example of repulsive potential for the case of three neighboring ABSs, marked by the black crosses.

The ABS stays in this state until the separation is not needed anymore. The transition $T2$ defines when the ABS leaves the separation state by using the simple interference measure:

$$T2 : S_{UE} \geq S_I + D, \tag{4.6}$$

where S_{UE} is the highest downlink RSS of connected UEs, S_I is the highest RSS of neighboring ABSs and D is the separation threshold in dBm.

4.2.1.3 Placement State

In this state, the locally optimal placement needs to be found. This state assumes that ABSs are well-separated. To find the location for each ABS, one of the centralized algorithms discussed in the previous chapter is used. The chosen algorithm uses the locations of locally known UEs as an input and therefore, solves the local problem only. This leads to a significant decrease of the computational time needed to find a solution.

The ABS stays in this state until the separation is needed again. The transition $T3$ defines when the ABS leaves the placement state by using the simple interference measure:

$$T3 : S_{UE} < S_I + D, \tag{4.7}$$

where S_{UE} is the highest downlink RSS of connected UEs, S_I is the highest RSS of neighboring ABSs and D is the separation threshold in dBm.

The complete AIDA algorithm is shown in Alg. 3. The AIDA algorithm uses the local UE locations u_l, the neighboring ABS locations s_l, the number of available RBs r and the centralized algorithm T used on a local subproblem as an input. As for T the following algorithms can be used: the Random Search, the Nelder-Mead simplex, the K-Means clustering, the PSO, the proposed PC and the proposed PC-Quality algorithms.

In line 3 we initialize the vector W to the current agent location. D is set to initial threshold value for the state transitions $T1$ and $T2$.

Consequently, the algorithm estimates the current state. The corresponding conditions are defined in lines 5, 7 and 14. If the agent has no neighboring ABSs or connected UEs (line 5), it executes the Random Walk algorithm in line 6 to explore the environment. When there is connected UEs or the ABS senses neighboring ABSs, it proceeds either in separation or placement state, depending on the interference measure as defined in lines 7 and 14 correspondingly.

In case if interference measure is above the threshold D, it calculates the repulsion force from neighboring ABSs using the artificial field method in line 10. When the force vector is calculated, the agent combines this vector with its current position W to separate itself from neighboring ABSs in line 12.

1: **procedure** AIDA
2: **Inputs:** local UE locations u_l (size n_l, $n_l \leq n$),
neighboring ABS locations s_l (size m_l, $m_l \leq m$),
number of available RBs r,
centralized algorithm T
3: $W \leftarrow$ current agent location, $D \leftarrow$ initial threshold value
/* *Initialization* */
4:
5: **if** $n_l = 0$ and $m_l = 0$ **then**
6: $W = RandomWalk()$
/* *Exploration* */
7: **else if** $m_l > 0$ or $S_{UE} < S_I + D$ **then**
8: $F \leftarrow (0, 0)$
9: **for** $i \leftarrow 1 \ldots m_l$ **do**
10: $F \leftarrow F + CalculateForce(s_{li})$
11: **end for**
12: $W \leftarrow W + F$
/* *Separation: sum up individual forces over all ABSs in neighborhood* */
13:
14: **else if** $n_l > 0$ or $S_{UE} \geq S_I + D$ **then**
15: $W \leftarrow CA(u_l, r, T)$
/* *Centralized algorithm for the placement on the local problem* */
16: **end if**
17: **return** W
/* *Return new way point* */
18: **end procedure**

Algorithm 3. – AIDA Algorithm

In case if interference measure is below the threshold, the agent calculates in line 15 the best position to serve UEs and to maximize its own capacity using the one of centralized algorithms discussed in Chapter 3.

Finally, the next location (waypoint) for the agent is returned as output (line 17).

In such a system, there is no single point of failure and the system tries to converge again if one agent is lost. Moreover, the complexity is reduced because each agent solves only a relaxed and smaller sub-problem, defined for a single ABS:

- The Random Search: each sample solution is a two-dimensional playground location of a single ABS (agent).

- The Nelder-Mead simplex: a single ABS in two-dimensional space has two degrees of freedom, thus the simplex has three vertices.

- The K-Means clustering: special case for $k = 1$, where the solution consists of a single centroid for local UEs coordinates. The centroid is the arithmetic mean of UEs positions.
- The PSO: each sample solution (particle) is a location of a single ABS (agent) in two-dimensional playground.
- The proposed PC and PC-Quality: these algorithms evaluate multiple possible clustering solutions on the capacity-based model. However, the complexity of evaluations is reduced.

4.2.2 Threshold for State Transitions

Previously we have defined two transitions $T2$ and $T3$ based on the interference measure, which uses the separation threshold D. Good value of the threshold D varies for different scenarios due to the different number of UEs and their distributions. In the case of multiple dense UE clusters, it is preferable to increase the threshold value D to achieve the separation of the ABSs between the clusters. In the case of one large sparse UE cluster, it is better to reduce the distance D and avoid placing the ABSs outside of the cluster. Also, if the number of ABSs increases, the downlink interference will add up and the optimal value of the threshold D will be different. Since we use only local information to make decisions on ABSs, it is not possible to easily synthesize a common model that would determine the optimal distance.

Therefore, in this case, we propose a Reinforcement Learning (RL) model-free approach that allows the ABS to learn the appropriate value of D.

4.2.2.1 Q-Learning to Calculate Separation Threshold

Q-learning, one of the methods for RL, is the very common learning applied for the robot movement [77]. It uses the temporal difference method and solves learning problems without models.

The problem of defining the separation threshold D can be viewed as a Finite Markov Decision Process (FMDP) with the 4-tuple $(\mathcal{Z}, \mathcal{A}, \Pi, \Upsilon)$, where $\mathcal{Z}$ and $\mathcal{A}$ are respectively the sets of all possible states and actions, Π are the transition probabilities between $\mathcal{Z}$ and $\mathcal{A}$ and Υ are the immediate rewards for the effects of completed actions. The separation threshold D is discretized into multiple states $\mathcal{Z}$, where each state is a particular value of the threshold and is referred to as z_t, and the set of actions $\mathcal{A}$ corresponds to transitions from one state to another. The immediate rewards Υ are defined through the capacity, achieved after taking a particular action. However, the transition probabilities Π are not known.

The goal of Q-Learning is to learn the policy of selecting actions π, which maximizes the expected value of the total reward over all successive steps. For this task, Q-Learning maintains the Q-table, which defines the expected reward value of each action in each state.

Before learning begins, Q is initialized to a possibly arbitrary fixed value. Then, at each time t the agent selects an action a_t, observes a reward υ_t, enters a new state z_{t+1} and Q-table is updated. The core of the algorithm is a simple value iteration update, using the weighted average of the old value and the new information:

$$Q(z_{t+1}, a_{t+1}) \leftarrow (1 - \Omega) \cdot Q(z_t, a_t) + \Omega \cdot (\upsilon + \delta \cdot \max_{a_{t+1}} Q(z_{t+1}, a_{t+1})) \tag{4.8}$$

where z_t is the current state, z_{t+1} is the new state, $Q(z_t, a_t)$ is the learned objective value, $\max_{a_{t+1}} Q(z_{t+1}, a_{t+1})$ is the estimate of optimal future value, Ω is the learning rate, υ is the reward and δ is the discount factor.

The learning rate Ω determines the balance between new information and previous knowledge. A factor of 0 makes the agent learn nothing (exploiting prior knowledge), while a factor of 1 makes the agent consider only the most recent information (ignoring prior knowledge).

The discount factor δ defines the importance of future rewards. A factor of 0 makes the agent short-sighted (considering only current rewards), while a factor of 1 orients the agent to the long-term reward.

In the dynamic scenario, ABSs should balance between new and historical information, and thus we have selected $\Omega = 0.5$ and $\delta = 0.5$.

4.2.2.2 Exploration Strategy in Q-Learning

The Q-Learning method does not specify which actions should be taken in each state during the process of updating Q-table. To find the optimal policy π, the agent needs to try each state-action pair many times, which implies the long learning process.

For many real-time tasks, the compromise between exploration and exploitation can be found and the literature suggests multiple strategies. In this thesis, we use one of the most common and simple strategies: the ϵ-greedy exploration strategy.

In the ϵ-greedy exploration strategy, the agent chooses at each iteration i to take either a random action a with a probability ϵ or the optimal action $a_{t+1} = \pi(z_{t+1})$ with a probability $1 - \epsilon$. We apply a naive algorithm, where ϵ is reduced on the constant decay factor φ at each iteration. In this work, we have selected $\epsilon = 0.5$ and $\varphi = 0.99$ which represent a moderate greedy strategy while allowing good learning results.

4.2.3 Discussion of the Proposed SO framework

As discussed previously, the proposed SO framework consists of multiple non-cooperating agents. Each ABS in the network is a self-containing AIDA agent and its decisions do not depend on decisions of other agents. The agents have no explicit communication among each other and interact only implicitly through the common environment. Thus, each agent learns how to switch between previously described control rules separately and relies only on its own perception of the environment.

When the agent comes close to the neighboring agent, it increases downlink interference at UEs. As the result, it strongly affects the plan of neighboring agent, which aims to maximize its own aggregated capacity. In this case, the system becomes *tightly coupled* and agents may frequently change their plans, which negatively affects the overall system stability [78]. The *separation* state in each agent is designed to prevent this behavior without a need for explicit communication between agents. The final goal of the *separation* state is to split up agents so that their actions only weakly and infrequently affect the plans of other agents (*weakly coupled system*).

The success of the proposed AIDA algorithm relies on the accuracy of the Q-Learning process and the correct balance between the exploration of new knowledge and exploitation of already existing knowledge. While we have done our best to ensure the correct learning balance, the learning process is a complex task and deserves further studies.

The AIDA algorithm exposes some instabilities, in particular in the following situations:

1. Potential force equilibrium: one of ABSs is symmetrically surrounded by other ABSs (the reader can imagine a circle of ABSs with a single ABS in the center of circle). Unless there is a decision of any ABS to break this equilibrium (for example, due to exploration in Q-learning), the ABS in the center experiences the force equilibrium, stays 'trapped' and creates unwanted interference;

2. 'Ping-pong' effects: high number of ABSs are located in small area and UEs are grouped in a few dense clusters. Because the ABSs are driven by their own selfishness (e.g. increase of the own capacity), they will not leave UE clusters (given that there is enough load) and will conflict with each other, creating strong downlink interference. This leads to the degradation of aggregated system capacity.

Another limitation results from *exploration* state of the AIDA algorithm. If UEs are separated by very large distances, the agents have slow convergence rates due to the stochastic random walk nature. However, this should not be a major problem because in traffic offloading scenario, the network operator should have locations of overloaded cells and can pass this information directly to ABSs before the mission begins.

While the AIDA algorithm is not perfect and can be improved, it still shows that the reasonable results can be achieved without global knowledge.

4.3 Simulation-based Evaluation

As mentioned previously, we can use different centralized algorithms during the *Placement* state in the proposed SO framework. In order to distinguish between purely centralized algorithms and algorithms used within the developed SO framework, we highlight the latter with prefix *SO* in front. In this section, we compare the previously discussed algorithms and examine their performance for different user distributions: we vary the user density and the number of user clusters.

4.3.1 Scenario

256 UEs were placed in the area of 2000 x 2000 m^2. The goal was to find a close-to-optimal placement, which maximizes the aggregated system capacity. Six algorithms were compared in the developed framework:

- the SO Random Search,
- the SO Nelder-Mead simplex,
- the SO K-Means clustering,
- the SO PSO,
- the proposed SO PC and
- the proposed SO PC-Quality algorithms.

We have selected the same configurations of algorithms as in Section 3.4.1. The parameter μ for the artificial force model is set to 0.01 to force the ABS to separate from neighboring ABSs for the whole playground. In the Q-Learning algorithm, 40 states with values from 1 to 40 are defined for the the separation threshold D. According to our experiments, this number of states should provide a reasonable approximation for the separation threshold. Algorithm parameters are summarized in Table 4.1. User locations are given and fixed during the simulation run, while the ABSs are initially located in the center of the area. The transmission power of ABSs is constant. The r-best-CQI resource scheduler ($r = 6$) is used.

Table 4.1. – Main algorithm parameters for the SO framework

Algorithm	**Parameter Values**
Artificial force field	$\mu = 0.01$
Q-Learning	$len(z_t) = 40$, $z_{tinit} = 1$ $z_{tstep} = 1$ $\Omega = 0.5$, $\delta = 0.5$ $\epsilon = 0.5$, $\varphi = 0.99$
SO Random Search	$t_{max} = t_{PC}$
SO Nelder-Mead, Simplex	$x_{atol} = 0.0001$ $f_{atol} = 0.0001$ $max_{iter} = 600$
SO K-Means	$k = 1$ (special case: arithmetic mean of local UEs coordinates)
SO PSO	$\psi_1 = 0.5$, $\psi_2 = 2.0$ $\phi = 0$, $t_{max} = t_{pc}$ $swarm_{size} = 8$ init distribution: $'random\ uniform'$

4.3.2 Analysis of Aggregated Capacity and Time Complexity

We analyze the influence of different problem parameters on the aggregated system capacity and the CPU time.

4.3.2.1 Impact of User Distribution on Aggregated Capacity

In order to analyze the impact of the user density on the algorithms performance, in this scenario we create 9 dense user clusters and vary their density α as described in Section 2.2.4.1. The results for the average aggregated capacity for various densities and 8 ABSs are presented in Fig. 4.6.

Most SO algorithms show a fluctuating behavior, except for the stochastic SO PSO and SO Random Search algorithms. These variations are the result of partial knowledge of the system and local decision making. The SO K-Means and the SO Nelder-Mead show sub-optimal results in terms of the aggregated system capacity. The proposed SO PC and SO PC-Quality show consistently good results, without a significant difference between them. In contrast to the SO K-Means and the SO Nelder-Mead, the proposed algorithm evaluates multiple possible subsets of UEs and can oversee better subsets of UEs in terms of aggregated capacity located farther away. The consistency of the SO PC and SO PC-Quality also helps to improve the learning process.

Compared to centralized algorithms, the relative difference between the different algorithms for the SO framework is smaller. In SO systems, the correct separation between agents plays a more important role than the algorithm used for placement.

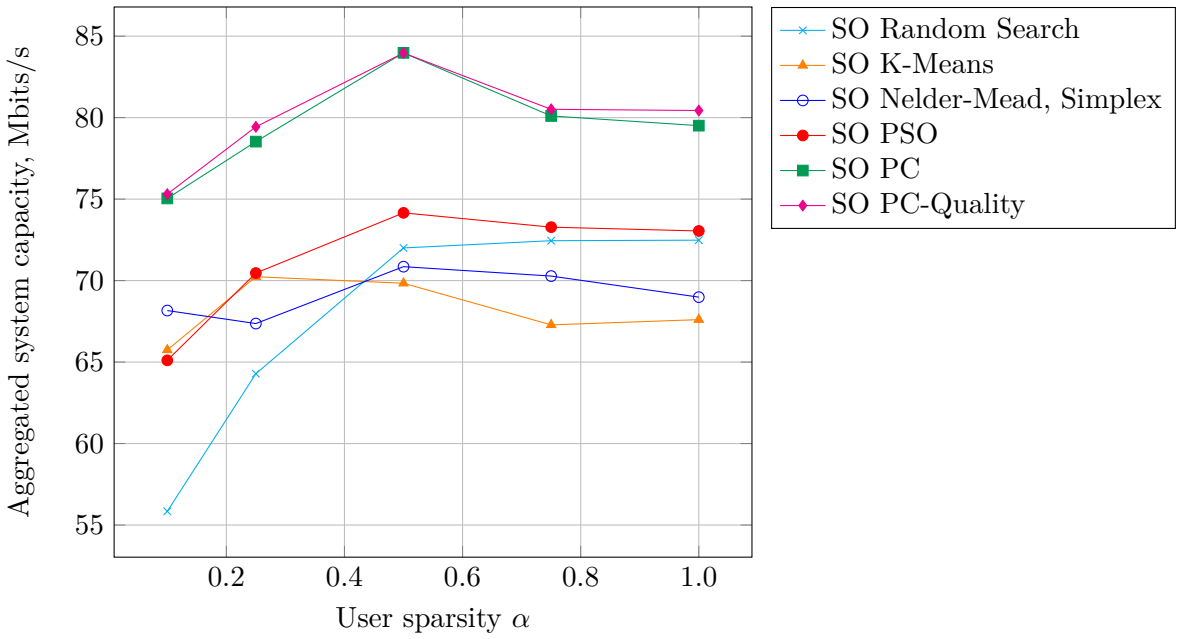


Figure 4.6. – Impact of user density on average aggregated network capacity for 8 ABSs and 256 UEs, $\gamma = 2.5$, 9 user clusters.

However, the algorithms still show similar performance for uniform user distribution, as it was for the centralized algorithms. Next, we fix the user density at $\alpha = 0.1$ to analyze the impact of changes in the number of user clusters on the performance of algorithms.

4.3.2.2 Impact of Number of User Clusters on Aggregated Capacity and CPU Time

In this scenario we fix the user density at $\alpha = 0.1$. The results for the average aggregated capacity vs. the CPU time for a variable number of user clusters and 8 ABSs are presented in Fig. 4.7. The corresponding Pareto front is shown in Fig. 4.8. The values in these figures depict the number of user clusters.

In comparison to the centralized approaches, the overall CPU time for the SO algorithms is significantly reduced (around $100x$). The SO PC-Quality algorithm has the biggest computational time, while the SO K-Means is the fastest algorithm as before. All algorithms have significantly lower performance in the case of 4 user clusters. The reason for this is the significant inter-cluster distance. As the distance grows, there is a greater chance that the SO framework will get into the *exploration* state and will randomly search for the users, what significantly reduces its efficiency.

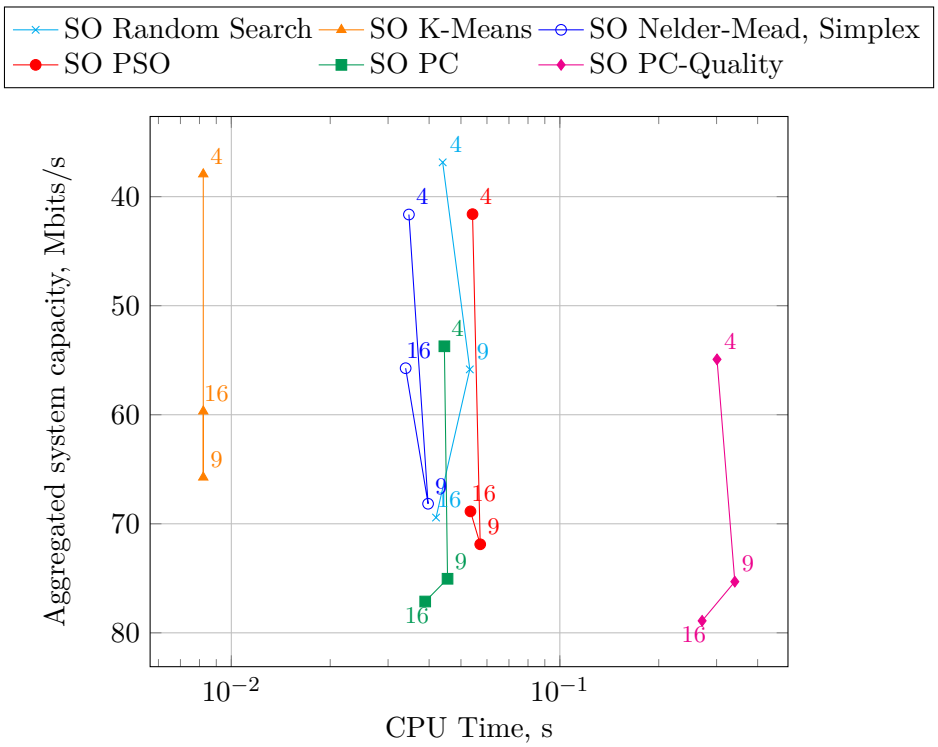


Figure 4.7. – Impact of number of user clusters on average aggregated network capacity and CPU time for 8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$.

For 16 user clusters we see a consistent performance for the PC and PC-Quality algorithms, which have only minor differences in the solution quality. The SO K-Means and the SO Nelder-Mead show sub-optimal behavior. The performance of the SO Random Search is not as poor as in the centralized case, but still worse than most of the algorithms. The separation algorithm positively influences results of the worst algorithms, limiting their search and excluding clearly sub-optimal solutions.

4.3.2.3 Centralized Algorithms vs. Self-Organized Framework

In order to compare the centralized and SO approaches, we have combined their results in Fig. 4.9. Here, the results for the average aggregated capacity vs. the CPU time for the variable number of user clusters and 8 ABSs are presented. The values in this figure depict the number of user clusters. We have chosen only the most efficient approaches for this comparison to simplify the figure. The summary of results and their relative comparison is given in Tables 4.2 and 4.3. Interested readers may find the comparison of all algorithms in the Appendix A.1 (Fig. A.2).

As expected, the centralized algorithms outperform the SO algorithms in terms of the aggregated capacity. This happens due to the partial knowledge of the SO agents.

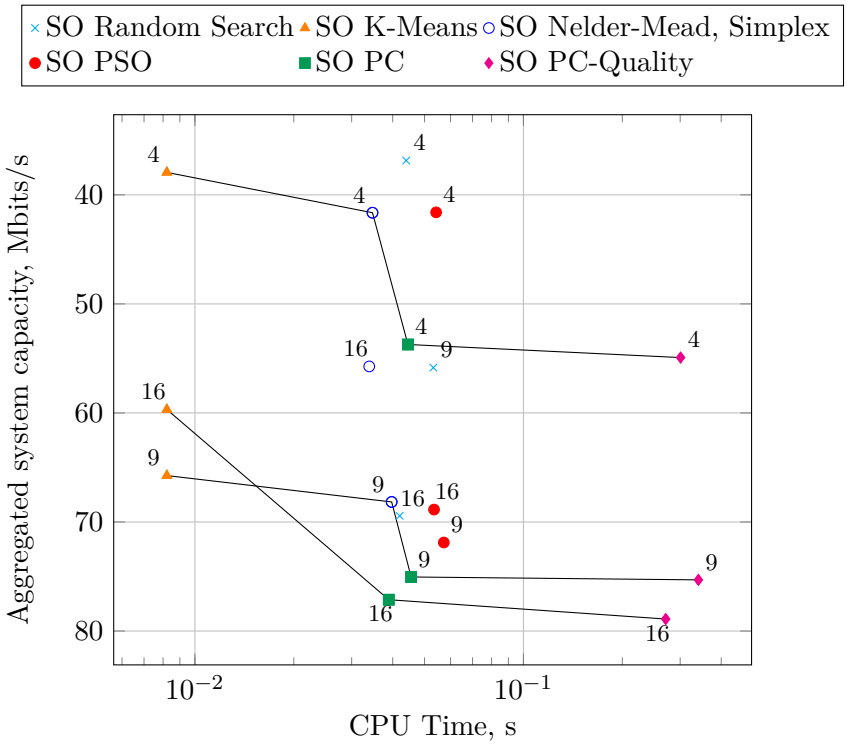


Figure 4.8. – Pareto front: impact of number of user clusters on average aggregated network capacity and CPU time (8 ABSs, 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$).

Table 4.2. – Absolute and relative comparison of aggregated capacity between algorithms for different amount of user clusters (comparison between centralized and SO algorithms, only the best algorithms). The results of the PC-Quality algorithm are used as a baseline for the relative comparison.

	Aggregated Capacity, *Mbits/s*			**Relative Loss in Aggregated Capacity,** %		
# clusters / **Algorithm**	4	9	16	4	9	16
PC	74.36	84.10	84.41	−2.61%	−1.27%	−2.45%
PC-Quality	76.35	85.18	86.21	0.0%	0.0%	0.0%
K-Means	69.84	78.26	49.74	−8.53%	−8.12%	−42.30%
SO PC	53.72	75.04	77.13	−29.64%	−11.90%	−10.53%
SO PC-Quality	54.92	75.31	78.90	−28.07%	−11.59%	−8.48%
SO K-Means	37.94	65.74	59.68	−50.31%	−22.82%	−30.77%

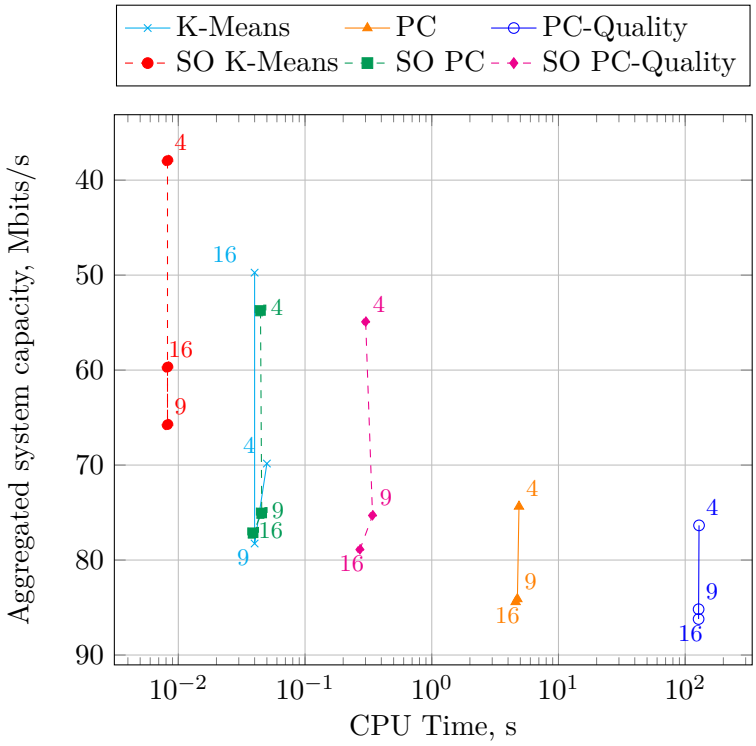


Figure 4.9. – Impact of number of user clusters on average aggregated network capacity and CPU time for 8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$ (comparison between centralized and SO algorithms, only the best algorithms).

Table 4.3. – Absolute and relative comparison of CPU time between algorithms for different amount of user clusters (comparison between centralized and SO algorithms, only the best algorithms). The results of the SO PC algorithm are used as a baseline for the relative comparison.

	CPU Time, *s*			**Relative Gain in Speed,** times		
# clusters / **Algorithm**	4	9	16	4	9	16
PC	4.88	4.74	4.59	122.00	94.80	114.75
PC-Quality	128.6	127.58	127.56	3215.0	2551.6	3189.0
K-Means	0.05	0.04	0.04	1.25	0.80	1.00
SO PC	0.04	0.05	0.04	1.00	1.00	1.00
SO PC-Quality	0.3	0.34	0.27	7.50	6.80	6.75
SO K-Means	0.01	0.01	0.01	0.25	0.20	0.25

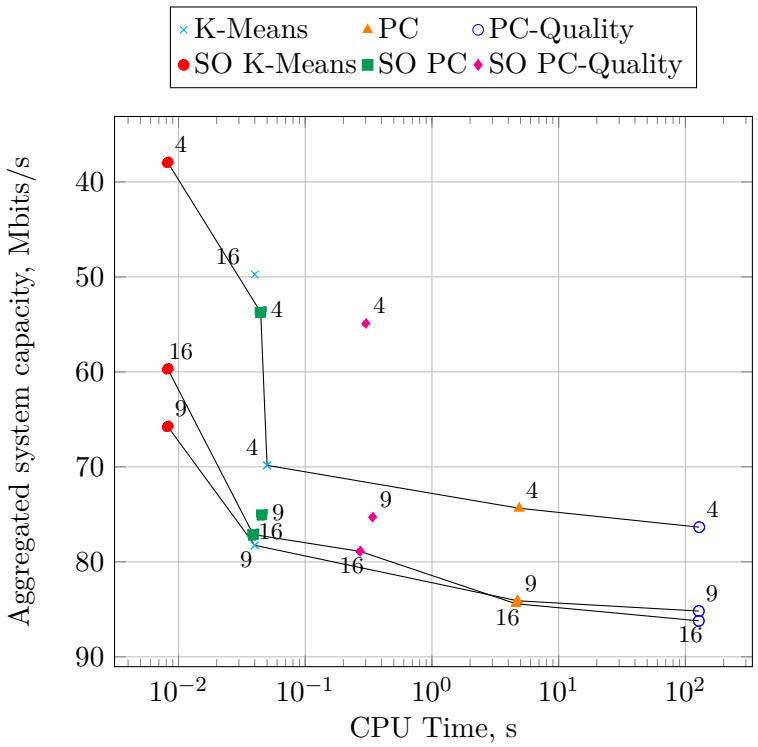


Figure 4.10. – Pareto front: impact of number of user clusters on average aggregated network capacity and CPU time (8 ABSs, 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$) (comparison between centralized and SO algorithms, only the best algorithms).

The performance of the SO K-Means has the most significant loss in aggregated capacity in comparison to the centralized PC-Quality algorithm: $\sim 50\%$, $\sim 23\%$ and $\sim 31\%$ for 4, 9 and 16 user clusters correspondingly.

The SO PC and SO PC-Quality show comparable results and have lower loss than the SO K-Means: the biggest loss in the aggregated capacity $\sim 30\%$ is observed for 4 user clusters, while for 9 and 16 user clusters the loss is only $\sim 10\%$. Better results in the case of complex user distributions with high amount of clusters can be explained by learning parameters, which do not favor long-term explorations.

The interesting insight is that the centralized K-Means algorithm achieves very similar results in terms of aggregated capacity for 9 user clusters when compared to the SO PC and SO PC-Quality algorithms. For 4 user clusters the SO PC performs significantly worse than the centralized K-Means ($\sim$ 30% worse), while for 16 user clusters the SO PC is able to outperform the centralized K-Means ($\sim$ 35% better). As the CPU time for the centralized K-Means and the SO PC algorithms is similar, the interesting observation can be made: centralized approaches, which have global knowledge, may actually perform worser or similarly than the SO approaches without global knowledge.

In our case, the reason is the approximation error of the centralized K-Means algorithm, discussed in Chapter 3. Another issue is the performance of non-cooperative SO agents e.g. how well the agents can separate themselves and reduce the influence on neighboring agents e.g. achieve a weakly coupled SO system.

As for the CPU time, the SO PC algorithm is approximately 100 times faster than the centralized PC algorithm. This difference is due to the distributed computation. Due to low absolute CPU time and scalability, the SO PC algorithm can be readily recommended for real-time scenarios.

4.3.2.4 Impact of User Mobility on Aggregated Capacity

In this scenario, we analyze the impact of user mobility on aggregated system capacity for centralized and SO algorithms. The mobility of UEs is modeled using RPGMM mobility model (more details in Section 2.2.4). In this model, each user belongs to a group (cluster) and follows the logical center of the group (leader). The leader defines the groups' motion behavior. We fix the user density at $\alpha = 0.1$, the path-loss exponent

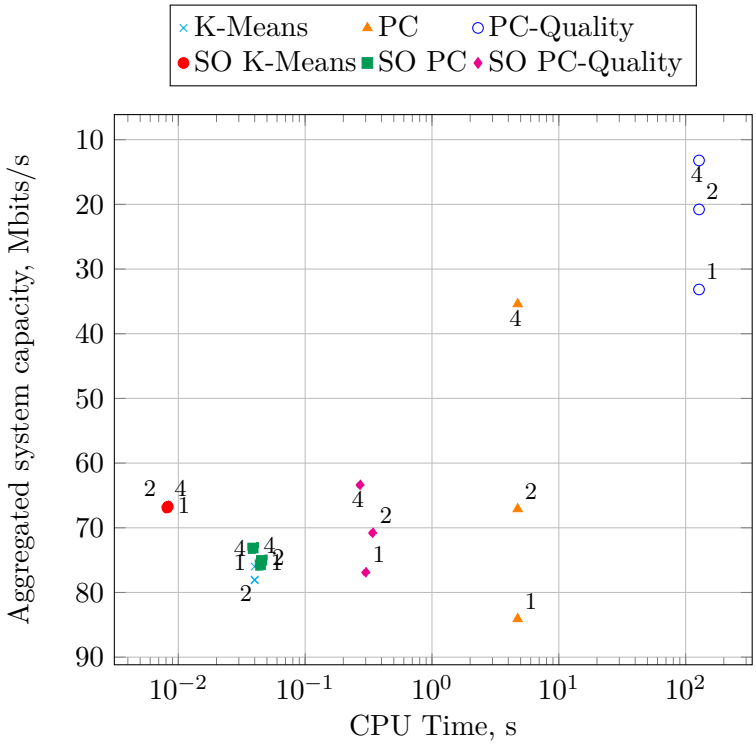


Figure 4.11. – Impact of UE speed (1, 2 and 4 m/s) on average aggregated network capacity for 9 mobile user clusters, 8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$ (comparison between centralized and SO algorithms, only the best algorithms).

at $\gamma = 2.5$ and the number of user clusters at 9. The results for the average aggregated capacity for different user speeds are presented in Fig. 4.11.

It is evident that the execution time of the placement algorithm affects the average aggregated capacity. While the algorithm is running, users continue to move and the solution of the algorithm becomes obsolete. This is especially true for highly mobile UEs and for the computationally expensive algorithms. The centralized PC and PC-Quality algorithms cope not well with higher UE speeds because of the inability to quickly recalculate a new solution. At the same time, for the centralized K-Means, the SO PC, the SO PC-Quality and the SO K-Means algorithms, we see almost constant results for different UE speeds. This shows the ability to follow the user cluster by quickly providing a new solution for the placement problem. This clearly shows the importance of scalable algorithms for placement problems in dynamic scenarios.

4.4 Conclusion

In this chapter we have discussed how distributed multi-agent systems can improve reliability and scalability in aerial base station deployment. We have introduced a novel distributed self-organized framework AIDA, where non-cooperative agents employ only local information to decide upon their placement.

During devising of the AIDA algorithm, important design issues of centralized and distributed systems were discussed, especially tight and weak system coupling. Agents in the proposed SO algorithm use deliberative decision making to switch between multiple reactive control rules. These control rules are defined for three basic situation (states): exploration, separation and placement.

In the separation state, the agent uses the artificial potential model to separate ABSs from each other and ensure a weakly coupled SO system. In the weakly coupled SO system, agents affect decision of other agents only infrequently and insignificantly. This leads to a better local accuracy for each agent. In the placement state, the developed centralized algorithms are executed using local input.

In order to find the proper separation distance locally and to switch between separation and placement states, RL algorithm, Q-Learning, is used in each agent independently. Because each agent needs to learn only one parameter, the learning state space is reduced, as well as the learning time.

Simulation results show that the proposed SO PC algorithm achieves nearly the same solution quality as the centralized PC version (around 10% less capacity), while significantly reducing the computational and communication needs. The computational time is reduced around 100 times.

In our opinion, the AIDA shows a strong potential for applications in ABS placement problems in temporary-event scenarios as a computationally-efficient, fault-tolerant and scalable solution, which requires only local information. However, the proposed SO algorithm shows deviations in the solution quality because of local knowledge and non-cooperative agent nature. It also is prone to instabilities, created by force equilibriums and conflicting agent objectives in presumably low number of situations. These issues deserve further studies. The proposed SO algorithm has few soft spots, there is a room for further enhancement and development.

Chapter 5

Experimental Model Validation

> If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.
>
> John von Neumann

In previous chapters, we have analyzed the ABS placement problem and compared multiple placement approaches using computer simulation. However, a successful application of computer simulation depends on the validity and accuracy of the applied simulation model. In particular, the A2G radio propagation model is important for our scenario. As we described in Section 2.2.2, there are three A2G models proposed in the literature for our scenario. These models have not yet been validated in real experiments.

This situation ultimately motivates us to conduct the real-world experiment based on the developed ABS prototype, to choose and validate the most suitable A2G model for the scenario in this thesis. In this chapter, we answer the following questions based on the real-world experiment:

- which model is the most suitable and accurate for our scenario?
- how well can the model perform prediction for different propagation environments?

As the next step, we present the developed experimental testbed.

5.1 Aerial Base Station Hardware Design

New advances in technologies brought us cheap, small and powerful Unmanned Aerial Vehicle (UAV) platforms. According to our investigations and real-world experiments

in [3], Vertical Take-Off and Landing (VTOL) vehicles, such as Micro-Aerial Vehicles (MAVs), are the most suitable for our scenario due to its ability to hold the position with high accuracy. They are also inexpensive and have a reasonable flight time for short missions.

To implement base station functionality, we have used a Software-Defined Radio (SDR). The type of SDR used for this project is bladeRF x40 made by Nuand, California [79], which has a small form factor (12.7×8.89 cm^2) and weight (80 g). It also supports full-duplex communication. The type of antennas used in this work is DeLock 88416. It is a miniature antenna (137 mm length) and provides operation at Global System for Mobile communications (GSM), Universal Mobile Telecommunications System (UMTS) and LTE bands (850/900/1800/1900 MHz) with $1 \sim 3.5$ dBi gain.

The assembled prototype, which we call Software-defined radio enabled micro AerIal vehicLe (SkySAIL), is shown in Fig. 5.1 and its main technical characteristics are summarized in Table 5.1. A more detailed description of the SkySAIL can be found in Appendix A.3.

As the next step, we describe the software, which we have used during the experimental model validation.

5.2 Software Design

To perform the real-world experiment, the appropriate software is needed to realize a BS functionality. Unfortunately, at the time of writing this thesis, there was no stable software for LTE for ARM-based devices. Instead, we have implemented a fully functional GSM BS using the open-source software OpenBTS with minor modifications [80].

OpenBTS has an interface for an SDR in order to provide a GSM air interface ("Um" interface) for standard GSM handsets. Technically, the OpenBTS replaces the original GSM core network infrastructure from the network layer upwards. Instead of forwarding

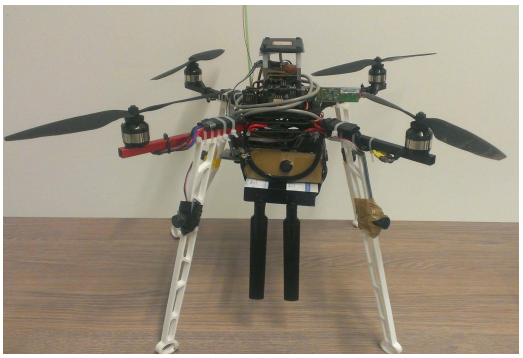


Figure 5.1. – SkySAIL platform.

Table 5.1. – Technical parameters of SkySAIL.

Parameter	**Value/Name**
CPU	1.7 GHz Quad-Core processor
Random Access Memory (RAM)	2 GByte
Universal Serial Bus (USB)	3 x High speed USB 2.0 Host ports
SDR type	Nuand BladeRF
Frequency range	300 MHz - 3.8 Ghz
Transmission power	+6 dBm
Power supply	4 cell LiPo batteries (3300 mAh)
Flight time	Up to 24 minutes
Installed Operating System (OS)	Gentoo Linux
Size	64x64x36 cm
Weight	1.3 kg including battery

call traffic to operators' mobile switching center, the OpenBTS delivers calls via Session Initiation Protocol (SIP) to a Voice-Over-IP (VoIP) Private Branch Exchange (PBX) switching software, such as Asterisk [81]. Asterisk can be installed on the same Linux computer forming a self-contained cellular network, which is illustrated in Fig. 5.2.

5.2.1 Measurement Approach

To obtain RSS measurements, the specific GSM procedures are used, which are explained below.

A GSM UE can be in two different Radio Resource Control (RRC) states: *Idle* and *Dedicated*. When a UE is turned on, it is in the *Idle* mode, until it receives a call or initiates a call, or receives Short Message Service (SMS), or transmits General Packet Radio Services (GPRS) data. In the *Idle* mode, UE will continue monitoring downlink signal strength of serving cell and neighbor cells to ensure it is attached to the best

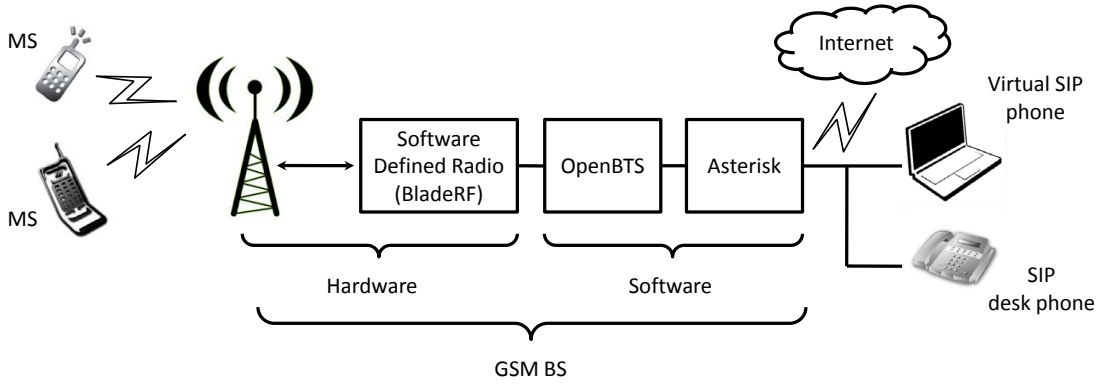


Figure 5.2. – OpenBTS-based GSM network structure.

available cell. However, the UE does not send any measurements to the serving BS. In order to allow the UE to transmit measurements to its serving BS, the UE has to be in the *Dedicated* mode.

As soon as the RRC connection is established, the GSM UE changes its state to the *Dedicated* mode. While in the *Dedicated* mode, the UE is constantly monitoring and transmitting measurement reports to its serving BS on Slow Associated Control Channel (SACCH), to implement power control and to help the BS to decide which is the best cell to handoff to. The SACCH carries power control and timing information in the downlink direction (towards the UE), RSS (Received Level Downlink (RXLEV_DL)), link quality reports in the uplink direction (towards the BS) and International Mobile Subscriber Identity (IMSI), the unique identifier of the UE. The measurements are made over each SACCH multiframe, which consists of 104 TDMA frames (duration of 480 ms) for a Traffic Channel (TCH) and 102 TDMA frames (duration of 470.8 ms) for Standalone Dedicated Control Channel (SDCCH) [82].

The reported parameter (RXLEV_DL) is the averaged received signal level measurement in dBm, taken within the reporting period of one SACCH multi-frame length (duration of 480 ms).

To keep all the UEs in the *Dedicated* mode, we have used the SMS service. It was periodically used to send SMSs to the UEs from the GSM BS.

Periodic reception of SMS by the UEs ensures that they stay in the *Dedicated* mode during the measurement procedure. There is also another way to keep users in this mode – the use of a Short Message Service Cell Broadcast (SMSCB) for broadcasting SMS to all the UEs. Nevertheless, in this work Short Message Service Point-to-Point (SMPP) service was used, because most of the UEs are not configured to receive the SMSCB by default.

5.2.2 Data Collection Approach

To implement the designed measurement procedure, several functions were added into the existing source code of the OpenBTS.

The first function is responsible for maintaining the database which contains IMSIs of the connected UEs. When a new UE has established a connection to the BS, its IMSI has to be inserted into a database. It is necessary to know IMSIs of the connected UEs in order to send SMS and keep UEs in the *Dedicated* mode during the measurement procedure.

The second function implements the procedure of obtaining RXLEV_DL measurements from the UEs. The third function recurrently schedules SMS messages for delivery. These functions are responsible for the measurement process, which is shown in Fig. 5.3.

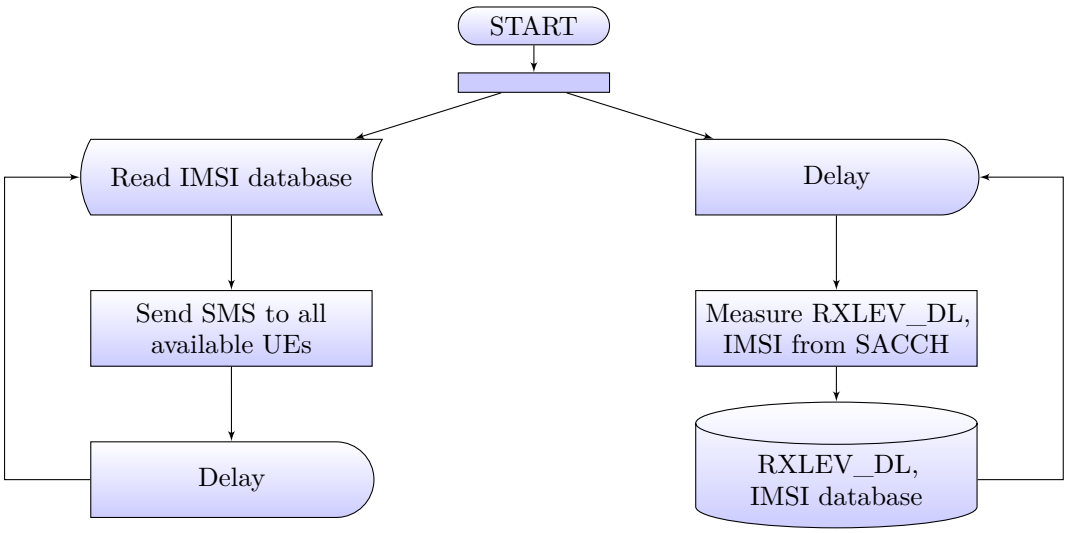


Figure 5.3. – Flowchart of the RXLEV_DL measurement procedure.

The BS reads the IMSI database and sends SMSs to all connected UEs, receives and stores their RXLEV_DL measurements in a database. Procedures of the SMS sending and obtaining RXLEV_DL measurements run simultaneously. However, before obtaining RXLEV_DL measurements, it is recommended to wait at least one second until the UEs receive SMS and change their states from the *Idle* to the *Dedicated* mode.

The RXLEV_DL measurements are combined with the IMSI of the UE and the reported Global Navigation Satellite System (GNSS) coordinates of the MAV and saved into the database. Obtained dataset is used for the model validation in the next section.

5.3 Experimental Validation of Air-to-Ground Propagation Models

In the following subsections, we present the experiment setup, preprocessing and evaluation of the obtained data and its comparison to the A2G propagation models introduced in Section 2.2.2.

5.3.1 Experiment Setup

An experimental evaluation of the propagation model was conducted at the Ilmenau University of Technology in Germany. For the experiment, we have chosen an area on the campus with a size of 58×156 m^2, shown in Fig. 5.4. This area represents two different environments on the small-scale level – the obstructed and the open environment. The obstructions were introduced by the "Zusebau" office building, which can be seen in

Fig. 5.4. Near this building, there is an open area, which was considered as the open environment. Surrounding buildings, moving cars and humans were present at the time of the experiment.

During the experiment, eight cellular phones (aka UEs) were placed in the chosen area. Different models of smartphones were used to introduce diversity. Three UEs (Motorola X Play, Apple iPhone 5s, Apple iPhone A1533) were placed at the open area. An Apple iPhone A1429 was located in the close vicinity of the building. Four other UEs (Motorola G3, Motorola XT910, Motorola G6 Plus and HTC One S) were located inside two internal courtyards of the building. UE locations are shown in Fig. 5.4 and the detailed information is summarized in Table 5.2.

SkySAIL was flying along the chosen trajectory (Fig. 5.4) with the speed of 3 m/s and receiving RSS measurements. These measurements along with the current SkySAIL's location and altitude were saved into the database (see Section 5.2). In the performed experiment, the default GSM channel combination V was used. With this channel combination, only four UEs can be served for sending SMS via one SDCCH. Three experiment repetitions were made on 30, 35 and 40 meters of altitude. The experiment site parameters and weather conditions are summarized in Table 5.3.

5.3.2 Experiment Results

To find the most suitable and accurate A2G model for our scenario, we compare the real measured data to the data generated by simulation models, in both quantitative and qualitative ways. As for quantitative methods, we make the comparison in terms of Root Mean Square Error (RMSE) and the R-squared metrics to show which model has better fitness.

Because the propagation models are generalized to predict different scenarios, we do not expect perfect data modeling. Rather, we compare the models in a qualitative

Table 5.2. – UE information: type, location and Global Positioning System (GPS) coordinates in decimal degrees

UE Model	Location	Latitude	Longitude
Motorola G6 Plus	Courtyard	50.6818520	10.9402803
Motorola G3	Courtyard	50.6819004	10.9403862
Motorola XT910	Courtyard	50.6816048	10.9403238
HTC One S	Courtyard	50.6816256	10.9401675
Apple Iphone A1429	Open Area	50.6822226	10.9401377
Motorola X Play	Open Area	50.6825784	10.9402659
Apple Iphone 5S	Open Area	50.6825302	10.9402865
Apple Iphone A1533	Open Area	50.6823509	10.9402996

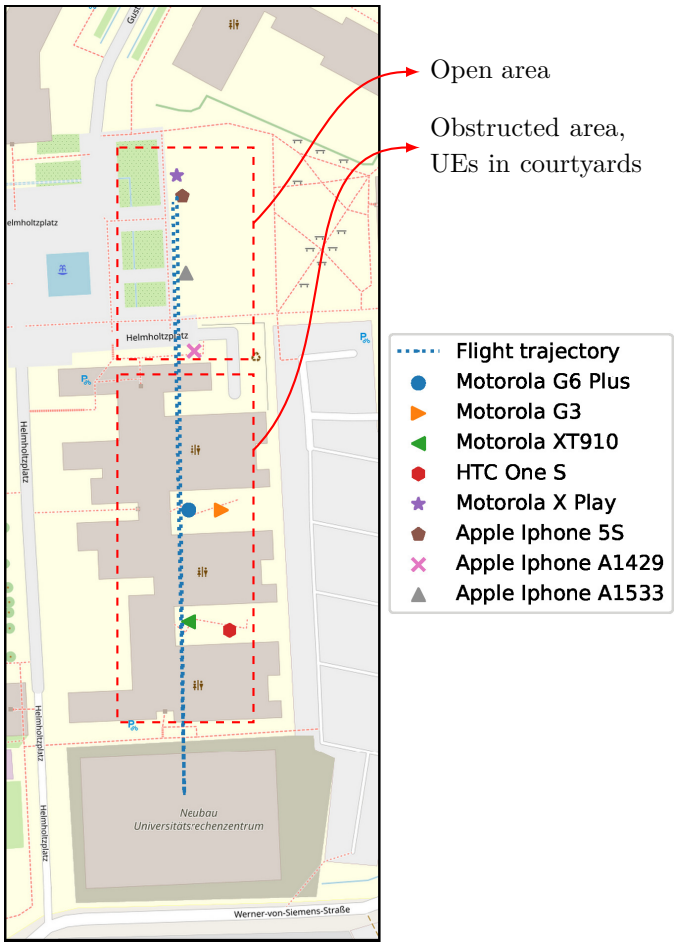


Figure 5.4. – Trajectory of the SkySAIL and UE locations during the real-world experiment. The map data is taken from [83].

Table 5.3. – Weather and experiment setup.

Parameter	**Value/Name**
Air temperature	24° C
Humidity	62, %
Speed of wind	20, km/h
Air pressure	1008, mb
Area size	58×156 m^2
Number of UEs	8
GSM Band	1800 MHz
TX power	+6 dBm
Experiment repetitions	3

way to show the relative model behavior. We compare two scenarios (UEs located at courtyards and at the open area) and show how the difference between them is modeled.

At the open area we expect to see lower deviations in RSS levels because of few obstructions in the propagation path in comparison to the courtyards of the building. Users, located in courtyards, are in close proximity with the building walls and thus, the LoS communication is less probable (except for elevation angles close to 90°). Also, these areas are expected to show different environment parameters, namely suburban and urban city parameters.

5.3.2.1 Data Processing

Prior to the analysis, the collected data need to be processed. Using the location data of UEs and the SkySAIL, elevation angles and distances at each trajectory point are calculated.

To calculate the distances, the coordinates should be first converted from GPS decimal degrees into the metric scale. This can be achieved using the haversine formula. This formula determines the great-circle distance (also called spherical distance) between two points on a sphere given their longitudes and latitudes:

$$d_s = 2R_e \cdot arcsin\left(\sqrt{sin^2\Big(\frac{\rho_2 - \rho_1}{2}\Big) + cos(\rho_1)cos(\rho_2)sin^2\Big(\frac{\chi_2 - \chi_1}{2}\Big)}\right), \tag{5.1}$$

where ρ_1, ρ_2 are latitudes, χ_1, χ_2 are longitudes and R_e is the radius of the sphere (we take the Earth radius, equal to 6371 km for the WGS84 ellipsoid). Using the distance d_s and the altitude of the SkySAIL, it is possible to calculate the elevation angle θ, as described in Section 2.2.2. Then, for each θ, the RSS values were generated using the A2G state-of-the-art models.

5.3.2.2 Evaluation Metrics

A well-fitting regression model results in predicted values close to the observed data values. In this thesis, we use two common metrics to understand the fitness of models: RMSE and R-squared.

The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the dataset and is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{K}(y_i - f_i)^2}{K}}, \tag{5.2}$$

where $y_1, \ldots, y_N$ are the values of the dataset, $f_1, \ldots, f_K$ are the predicted values by the model and K is the length of the dataset.

As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance and has the useful property of being in the same units as the response variable. The good model should have the RMSE close to or less than the standard deviation of the dataset [84].

R-squared is the proportion of the variance in the dependent variable that is predictable from the independent variable. It is the proportional improvement in prediction of the regression model in comparison to the mean model and is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{K}(y_i - f_i)^2}{\sum_{i=1}^{K}(y_i - \bar{y})^2}, \tag{5.3}$$

where $y_1, \ldots, y_K$ are the values of the dataset, $f_1, \ldots, f_K$ are the predicted values by the model and $\bar{y}$ is the mean of the dataset.

The R-squared metric can be interpreted as follows: when it is less than zero, it indicates that the proposed model makes poorer predictions over the mean model, when zero – that it does not improve prediction over the mean model, and R-squared equal to one indicates perfect prediction [85].

As the next step, we present a comparison between the real and the simulated data using the defined evaluation metrics.

5.3.2.3 Comparison of Air-to-Ground Models

In Fig. 5.5 the comparison between measured data and simulated data for all UEs is shown. It can be seen, that when the elevation angle increases, the RSS increases as well because of two main reasons: the distance between the SkySAIL and the UE is smaller and the probability of having LoS is higher. In our experiment, after about 40° the RSS stabilizes, because there are no obstructions between the SkySAIL and UEs. However, at smaller elevation angles, the RSS deviation increases due to shadowing effects (NLoS communication).

The model from [32] (green diamond marker in Fig. 5.5) almost neglects the impact of distance and thus, has the higher RSS values for LoS conditions (elevation angle > 30°).

The model from [29] (cyan triangle marker in Fig. 5.5) fits well with the measured data in terms of the average values when the elevation angle is less than 30°. However, the RSS deviation is very high on lower elevation angles, which does not corresponds to the real data. For the higher elevation angles, the RSS offset is very high, and after 60° the model does not correspond to the real data.

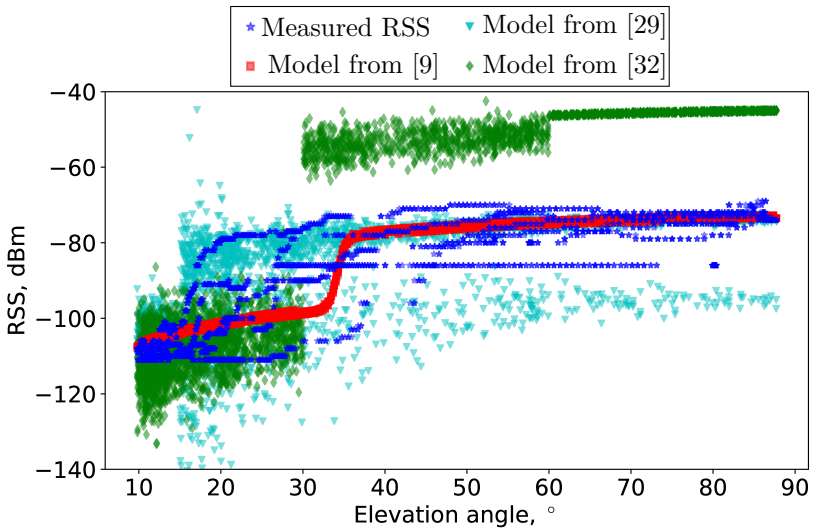


Figure 5.5. – Simulated and measured RSS values vs. elevation angle θ for state-of-the-art A2G models.

The model from [9] (red square marker in Fig. 5.5) shows the best results. At around 35° the transition from NLoS to LoS situation can be seen, which can be also observed from the data.

The quantitative data in terms of the RMSE and R-squared metric is presented in Table 5.4. The RMSE values near the standard deviation of the experimental dataset represent a good absolute model fit. The R-squared metric can be interpreted as follows: below zero or zero – no data correlation, 0.25 – weak data correlation, 0.50 – data is correlated, 0.75 – strong data correlation, 1.0 - perfect match.

As it can be seen from the data, the model from [32] provides the worst results and shows the worst RMSE value (22.01), which is bigger than the standard deviation of

Table 5.4. – The quantitative comparison of the real-world experiment data and state-of-the-art A2G models in terms of the RMSE and R-squared metric. The RMSE values near the standard deviation of the experimental dataset represent a good absolute model fit. The R-squared metric can be interpreted as follows: 0.25 – weak data correlation, 0.50 – data is correlated, 0.75 – strong data correlation.

	Model Performance		**Experimental Dataset**
Metric / **Model**	R^2	RMSE, dBm	Standard Deviation, dBm
Model from [9]	0.67	8.29	14.37
Model from [29]	0.06	13.94	
Model from [32]	−1.34	22.01	

the dataset (14.37). The R-squared metric is negative (−1.34), which means that there is no correlation between predicted and real data.

The situation is different for the model from [29]: according to the R-squared metric (0.06), there is only a minor improvement over the mean model. Meanwhile, its RMSE is less than the standard deviation of the dataset (13.94), while most of the values are close to the measured data points.

The model proposed in [9] shows the best results. R-squared is 0.67, which shows strong data correlation. As explained in the Section 2.2.2, this model uses a simplified calculation based on the ITU-T recommendation and can represent the city structure very accurately, which has a positive effect on the model performance.

The distribution of the RSS values is shown in Fig. 5.6. The model proposed in [9] closely matches NLoS and LoS propagation groups of the measured RSS, while other models show less accurate approximations.

Therefore, we conclude that quantitatively it is the most accurate model for our mixed scenario. As the next step, we analyze the model prediction behavior for this model in two different scenarios.

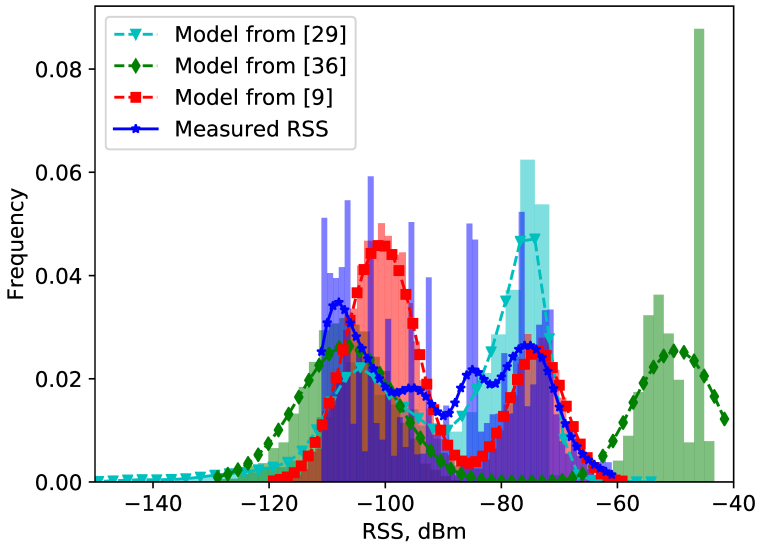


Figure 5.6. – Distribution of simulated and measured RSS values for state-of-the-art A2G models.

5.3.2.4 Scenario-Specific Propagation Models

In this section, we compare two different groups of users instead of the whole dataset (see Fig. 5.4): (1) the first group is located at courtyards, (2) the second group is located at the open area. Now we add a shadowing factor, which is defined as a random variable. Parameters obtained by meta-optimization are the following: for the courtyards ($a = 46, b = -2, \sigma = 7.0$), for the open area ($a = 30, b = -2, \sigma = 8.0$). The meta-optimization was done using grid search algorithm and the parameter space was defined as follows: $a \in [10, 90]$, step size equals to 1; $b \in [-90, 0]$, step size equals to 1; $\sigma \in [0, 25]$, step size equals to 1. To avoid overfitting, the 10-fold cross validation is used [86], where 80% of data is designated for the training dataset and 20% for the testing dataset.

Quantitative results are summarized in Table 5.5, and the comparison between measured data and predicted data by the chosen model for the both UEs groups is shown in Fig. 5.7 and 5.8.

It can be seen that the predicted data fit well with the measured data for both user groups. The prediction accuracy and data correlation is higher for the users located at the open area ($R^2 = 0.77$, RMSE= 7.29) in comparison to those at courtyards ($R^2 = 0.57$, RMSE = 7.29). Because of the higher data variance in the case of NLoS conditions and the limited number of samples, the model fit level decreases.

The data shows that the users, located at courtyards, start to observe NLoS effects at higher elevation angles ($< 46°$) than users at the open area ($< 30°$). This effect can be clearly seen in Fig. 5.7 and 5.8.

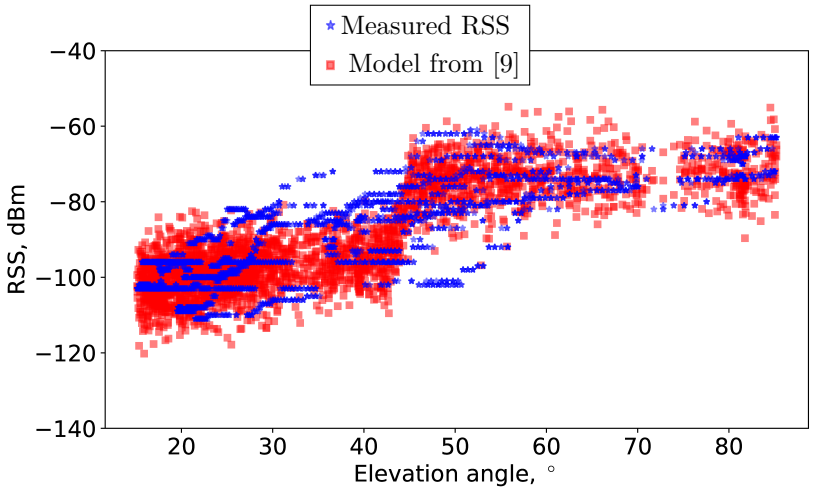


Figure 5.7. – Simulated and measured RSS values vs. elevation angle θ for the model from [9] at courtyards.

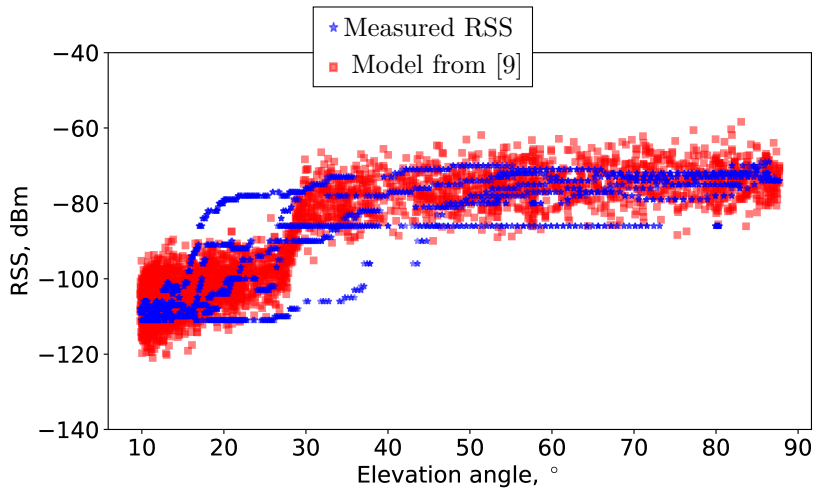


Figure 5.8. – Simulated and measured RSS values vs. elevation angle θ for the model from [9] at the open area.

Table 5.5. – The quantitative comparison of the real-world experiment data and the A2G model from [9] for different environment types in terms of the RMSE and R-squared metric. The RMSE values near the standard deviation of the experimental dataset represent a good absolute model fit. The R-squared metric can be interpreted as follows: 0.25 – weak data correlation, 0.50 – data is correlated, 0.75 – strong data correlation.

	Model Performance		**Experimental Dataset**
Metric / **Location**	R^2	RMSE, dBm	Standard Deviation, dBm
Courtyards ($a = 46, b = -2$)	0.57	8.50	12.96
Open Space ($a = 30, b = -2$)	0.77	7.29	15.11

This proves that the model can successfully predict RSS in different environments with high accuracy.

5.4 Conclusion

The main goal of this chapter was to choose and validate the A2G propagation model for computer simulation. In this chapter we have introduced the developed prototype for the ABS, called SkySAIL. This prototype is a multi-functional multi-rotor MAV equipped with a frequency-agile multichannel SDR. We have used the developed platform to perform the real-world experiment.

The best suitable A2G model for our scenario was chosen according to the experiment results based on the regression analysis with the help of RMSE and R^2 metrics. The most important findings can be summarized as follows:

- the most suitable model for our scenario was proposed in [9] and achieved the lowest RMSE (8.29 dBm) and the highest R-squared value (0.67) on the experiment data in comparison to other state-of-the-art models, which shows the good model prediction properties;
- the predicted data fits well to the measurements, when the two user groups (users located at building courtyards and at the open area) are examined separately, which proves the model's ability to distinguish between different environments (R-squared metric is 0.57 and 0.77 correspondingly). It was observed that the users in courtyards start to observe NLoS effects at higher elevation angles ($< 46°$) than users at the open area ($< 30°$).

Thus, we consider the analytical A2G propagation model from [9] as a valid and accurate simulation model in this thesis.

Chapter 6

Conclusion and Future Work

> A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it.
>
> Frank Herbert

The interest in aerial communication systems for traffic offloading from the existing cellular infrastructure has grown dramatically in recent years. These systems have great potential for dealing with dynamic and unpredictable scenarios such as disasters or temporary events, as they can adapt the positions of aerial base stations to user needs. In this thesis we have discussed the ABS placement problem, which requires exponential time to find the exact solution due to its complexity. We have proposed novel computationally efficient and effective solutions to meet the requirements of dynamic ABS placement in temporary-event scenarios.

6.1 Summary of Contributions

This thesis provides three major contributions to the field of aerial communication systems and its improvement. Briefly described, these contributions are as follows:

- We have formulated and analyzed the complexity of the aerial base station placement problem for traffic offloading scenarios. Based on the problem properties we developed the centralized hybrid algorithm Projected Clustering (PC), which analyzes the user distribution and uses a fast capacity approximation function. These contributions have been presented in Chapter 3.

- A novel SO framework AIDA for the ABS placement was proposed in Chapter 4. This framework relaxes the requirements for the global system state knowledge and reduces the computational and communication needs as well as increases the overall reliability of the system.

- To choose and validate the A2G propagation model for simulations, the experimental aerial base station prototype called SkySAIL was built on the basis of the quadrotor with software-defined radio on board. The design of SkySAIL and experimental results for the validation of state-of-the-art A2G models were presented in Chapter 5.

Besides, a custom continuous-time simulator has been developed for radio and mobility simulations. The object-oriented nature of the developed simulator can help other researchers to use the implemented models for their own purposes.

6.1.1 Centralized Aerial Base Station Placement

Our research has highlighted the importance of the optimal or near-optimal ABS placement for traffic offloading. We have formulated the corresponding optimization problem and showed that the problem is NP-hard and thus, requires heuristic optimization algorithms for relevant problem sizes.

The identified problem-specific knowledge in form of the relation of the received signal and distances between ABSs and UEs was used to approximate the original problem to a distance-based problem. We have proposed a novel hybrid algorithm PC, which quickly derives multiple solutions on the approximated distance-based model and evaluates them on the accurate SINR-based capacity model, validating the solution feasibility.

We have compared the proposed hybrid algorithm PC to the state-of-the-art approaches. Simulation results indicate that the proposed hybrid algorithm PC is able to solve the placement problem much faster than the other state-of-the-art algorithms while achieving the same or higher solution quality. For instance, in most of the scenarios, the PC is $\sim 89x$ faster and achieves $\sim 31\%$ higher system capacity than the Nelder-Mead algorithm (which generated the best solutions among other algorithms). On the other hand, the PC is $\sim 100x$ slower and achieves $\sim 41\%$ higher system capacity than the K-Means algorithms (the fastest algorithm available). In contrast to the state-of-the-art methods, our approach achieves stable near-optimal solutions by introducing only a minor increase in computational complexity in comparison to the K-Means algorithm. This ensures that our algorithm can deliver fast and accurate position updates to handle moving UEs.

In addition, the evidence from this study suggests that the user distribution plays a significant role for the ABS placement problem. For instance, many dense user clusters create numerous local minimums and the state-of-the-art algorithms perform sub-optimally ($\sim 52\%$ variance in the aggregated system capacity), while for the uniform user distributions the solution quality of different algorithms is similar ($\sim 10\%$ variance in the aggregated system capacity).

An implication of this study is that the utilization of the problem-specific knowledge can significantly improve the efficiency of the heuristic optimization.

6.1.2 Self-Organized Aerial Base Station Placement

The centralized placement algorithms solve the large-scale problems, assuming the availability of the global system state information. This leads to their limitations: first, the global system state information needs to be collected, introducing network overhead; second, they have a single point of failure; third, they have poor scalability, resulting in high absolute computational times (from minutes to hours) for relatively small problem sizes already.

We have proposed a novel self-organized framework AIDA to address these issues and relax the requirement of the global information availability. During the design of the AIDA algorithm, important design issues of centralized and distributed systems were discussed, especially tight and weak systems coupling. Agents in the proposed SO algorithm use deliberative decision making to switch between multiple reactive control rules. These control rules are defined for three basic situation (states): exploration, separation and placement. In the separation state, the agent uses the artificial force model to separate ABSs from each other and ensure a weakly coupled SO system. In the weakly coupled SO system, agents affect decision of other agents only weakly and infrequently. This leads to a better local accuracy for each agent. In the placement state, the developed centralized algorithms are executed using locally available information.

In order to find a proper separation distance locally and to switch between separation and placement states, RL algorithm, Q-Learning, is used in each agent independently. Because each agent needs to learn only one parameter, the learning state space is reduced, as well as the learning time.

Simulation results show that the proposed SO AIDA framework can achieve nearly the same solution quality as the centralized algorithms (around 10% less aggregated system capacity for 9 and 16 user clusters), while significantly reducing the computational (around $100x$) and communication needs. The results support the idea that SO approaches can be used in the large-scale ABS deployment.

However, it also is prone to instabilities, created by force equilibriums and conflicting agent objectives in presumably low number of situations. The proposed SO algorithm has few soft spots, there is a room for further studies and enhancement.

6.1.3 Experimental Validation of Radio Propagation Model

During modeling of the placement problem, we have found multiple state-of-the-art A2G propagation models for the low-altitude aerial platforms. However, they were derived analytically or empirically based on ray-tracing simulations and none of them was validated in real-world experiments. Thus, to choose the most suitable model, we have developed the prototype of the ABS, called SkySAIL, and have performed a real-world experiment.

The best suitable model for our scenario was chosen according to the experiment results based on the regression analysis with the help of RMSE and R^2 metrics. It was proposed in [9] and achieved the lowest RMSE (8.29 dBm) and the highest R-squared value (0.67) in comparison to the measured data, which shows good model prediction properties. Besides, the results indicate that this model can represent different environments with high degree of accuracy.

6.2 Future Work

The ABS placement is a new and attractive field for research. This thesis addresses only some important aspects, while leaving other ones for future work. For instance, our results show sub-optimal results for the SO framework in the cases of a small number of dense user clusters. This can be explained as an influence of the longer exploration phase due to limited system state information. Instead of a random walk as the exploration algorithm, a more sophisticated algorithm could be used. In particular, coordination mechanisms between multiple ABSs could be utilized for efficient space exploration.

The Q-Learning algorithm, used for separating the ABSs, deserves further investigation. Its parameters can be optimized, resulting in faster convergence rates and improved stability. Moreover, we have identified potential instabilities in the proposed SO framework, which should also be studied.

In this work, we have assumed to use the r-best-CQI scheduler. Depending on the optimization goal, this may not be the optimal choice. For example, if the goal is to ensure resource fairness, the stop criterion may be adapted.

ABS systems requires a backbone interface to communicate with the core network. We assume in this thesis that this interface is given. In reality, depending on the actual system design it may be an additional placement constraint. For example, in case the

ABS communicates with the ground station using a directional antenna, the placement should consider the distance between these entities and the ABS rotation angle. Another opportunity would be to use low-orbit satellites for this task.

The SO system, described in this thesis, can cope with the changes in the network and adapt to them. However, the replacement strategy is not defined explicitly. Depending on the actual design and UAV properties, it can be beneficial to include this knowledge in the system to achieve faster algorithm convergence. Moreover, the replacement history may include the best locations found previously to avoid unnecessary exploration phase.

So far scenarios with moving users have been considered as proof-of-concept and have not included advanced handover procedures. This can be studied in more detail, since it may be beneficial to not only make a handover to the next station as in traditional cellular networks but to allow the ABS to track a group of users continuously.

In this work we have studied the effects only on the downlink capacity. However, in some temporary-event scenarios users produce the content rather than consume it. This means that in some cases the uplink resource would be more important. Thus, the uplink channel can be investigated in more detail.

The A2G model can be affected by the Fresnel zone under some circumstances (as we observed during experiments, the RSS may vary when the ABS is very close to the ground).

Another important aspect is the accuracy of models, used by the ABS placement algorithms. If the environmental propagation parameters are not known, they need to be estimated first. Our initial results and the proposed estimation algorithm in [87] may serve as the basis for further investigations in this direction.

Appendix A

Supplementary Material

A.1 Additional Figures

The Zuzebau building, used in the experiment, consists of several courtyards. One of the courtyards is presented in Fig. A.1.

The comparison between all centralized and SO algorithms in terms of the aggregated network capacity for varying number of user clusters is shown in Fig. A.2.

Figure A.1. – The one of courtyards in the Zuzebau building at Ilmenau University of Technology.

Figure A.2. – Impact of number of user clusters on average aggregated network capacity and CPU time for 8 ABSs and 256 UEs, $\gamma = 2.5$, $\alpha = 0.1$ (comparison between centralized and SO algorithms, all algorithms).

A.2 Simulation Environment

In order to perform the radio and mobility simulations based on the proposed system model, a simulation environment is required. However, a generic simulation framework like SimPy [88] providing a full event-based simulation model is not necessary to perform the intended simulations. Instead, a custom simulator with a very small footprint is needed, supporting only the required functionality efficiently. Thus, a custom continuous-time simulator implemented in the Python programming language, called Aerial Base Station Simulator (ABSSim), has been developed for radio and mobility simulations.

A.2.1 Simulator Overview

The developed simulator has several features that simplify the development and analysis of algorithms:

- automated processing of multiple scenarios and multiple batches;
- support for parallel simulations;
- Graphical User Interface (GUI);
- automated collection and processing of statistics;
- connection interface for easy integration of new simulation models.

The general simulation flow is represented in Fig. A.3. First, the user specifies the simulation scenario and the parameters. The user can choose between interactive and non-interactive mode.

The interactive mode is designed for the development of new algorithms and online debugging of the process. In this mode, the user can monitor simulation progress through three types of GUI: *2D*, *3D* and *Unity* types. In the interactive mode it is also possible to manually control positions of users and ABSs. The non-interactive mode is designed to run multiple Monte Carlo simulations and to analyze multiple scenarios in an automated way.

To start the simulations, the user executes the *supervisor* process (*runfile.py*), which controls the simulation process. It creates N isolated processes using the source file *abssim_main.py*. N should correspond to the number of available CPU cores for the simulator. Each process works on a separate simulation run. At the end of a simulation run, the module *stats* collects the data from simulated entities, serializes it with the *pickle* module and stores it in the folder *results* with file extension **.obj*. Depending on the settings of the simulator, the serialized data can be archived on-the-fly with *gzip*. To

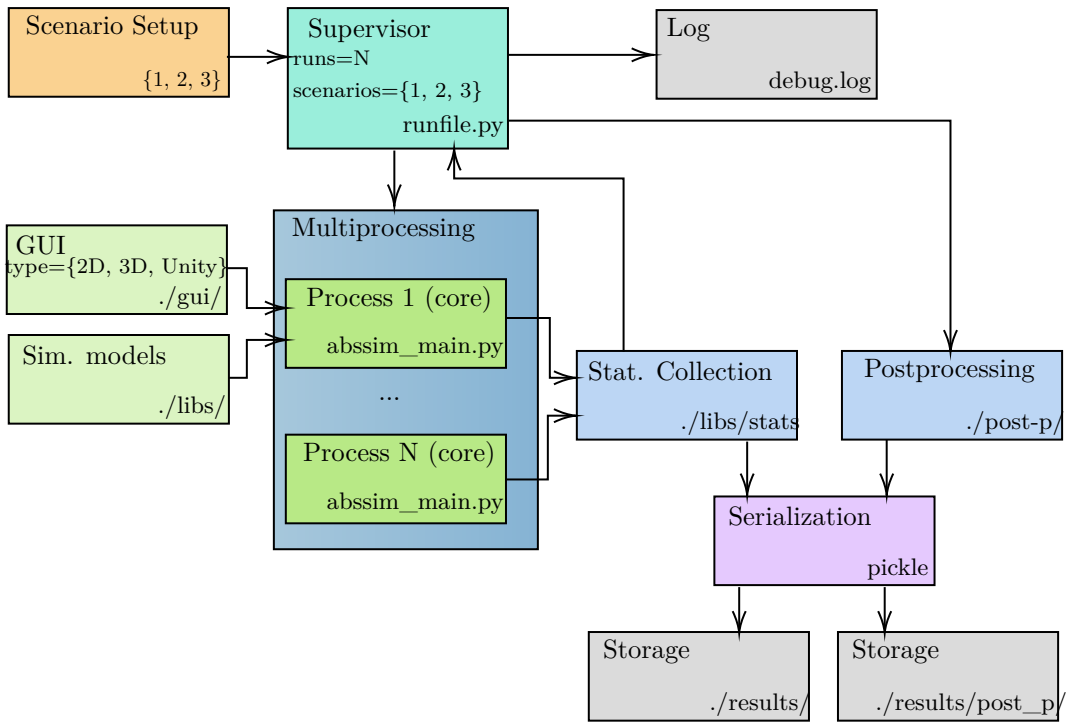


Figure A.3. – ABSSim simulation flow.

simplify the debugging process, several key metrics are reported back to the *supervisor* process: average aggregated system capacity, average convergence time, final positions of ABSs and UEs. The *supervisor* process repeats the procedure until all simulations are completed.

If required, the *supervisor* process can also post-process collected data with user-defined scripts in the *post-p* folder. The terminal output contains only limited information. A detailed log file is stored in the root directory (*debug.log*).

A.2.2 Single Simulation Run

A single simulation run is the core of the simulator. It is defined in the source file *abssim_main.py* and is coarsely presented in Fig. A.4. A single simulation consists of several entities that are represented by classes: *playground*, *actors* and *areas*. *Playground* defines the general simulation parameters. *Actors* represent simulated entities, e.g. UEs and ABSs. *Areas* represent variable radio environments.

At first, *playground*, *actors* and *areas* are initialized according to the simulation parameters and a *seed* for a Random Number Generator (RNG) is chosen. Then, the main loop is carried out until the maximum simulation time is reached. In the main loop, multiple procedures are executed iteratively: the position update, the radio update

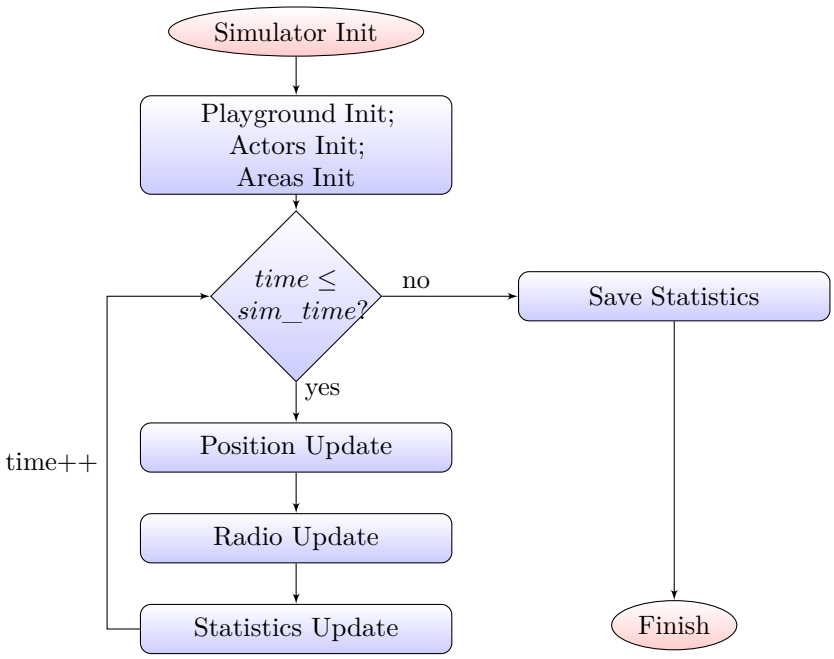


Figure A.4. – The single simulation run.

and the statistics update for each simulated entity. The *trajectory decision-maker* is integrated into the radio update procedure.

Finally, the statistical collection module is called to save the results, the simulation objects are destroyed and the supervisor process terminates the execution of the simulation run.

A.2.3 Simulator Structure

The simulator is designed using the object-oriented paradigm. Each object contains data in the form of attributes and code in the form of procedures or methods. The dependency graph for the ABSSim system level simulator is shown in Fig. A.5. The simulator consists of several main modules:

- **GUI** module interacts with the user at runtime. It displays current simulation progress and information about state of the simulation entities. The user can also change positions of entities, as well as some of their parameters for debugging purposes at runtime. The GUI module can be disabled to save resources during massive simulations.

- **Actors** module consists of classes for two main simulation entities, an UE and an ABS. In these classes, the behavioral logic of each unit is defined. This is

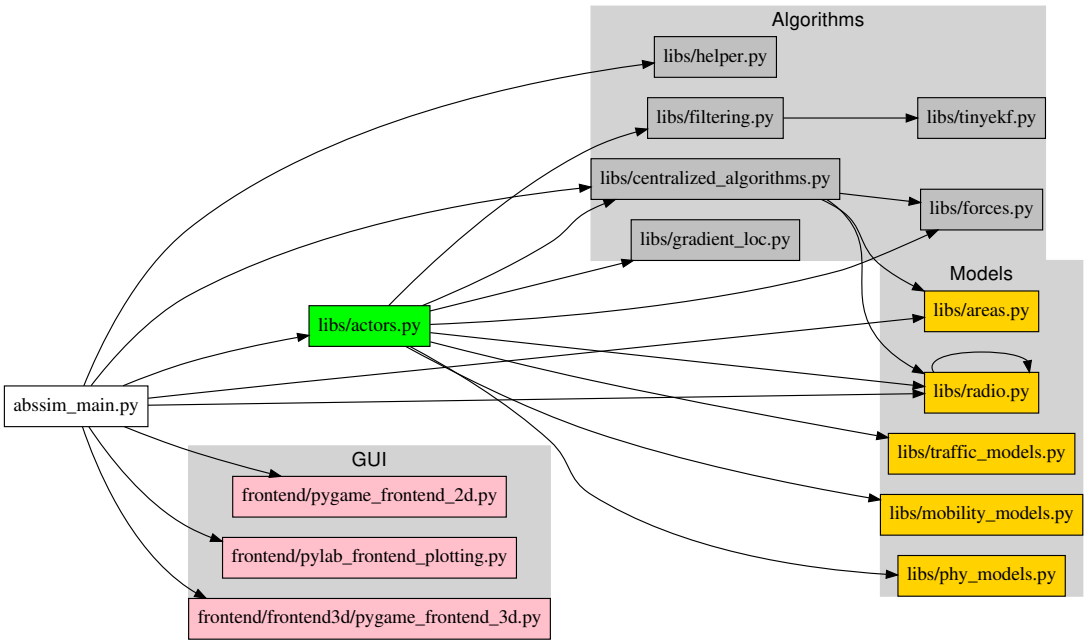


Figure A.5. – Dependency graph for the ABSSim system level simulator.

done with the help of the following libraries, which implement the corresponding models and algorithms:

- *radio.py*: radio propagation and channel models, as well as multiple scheduling strategies, described in Sections 2.2.1, 2.2.2 and 2.2.3;
- *traffic_models.py*: traffic models;
- *mobility_models.py*: mobility models, described in Section 2.2.4.2;
- *phy_models.py*: physical model of an aerial vehicle, described in Section 2.2.4.3;
- *areas.py*: radio environment parameters for user-defined areas, described in Section 2.2.2;
- *centralized_algorithms.py*: centralized optimization algorithms, described in Chapter 3;
- *forces.py*: artificial potential model, described in Chapter 4;
- *rl_agent.py*: the reinforcement learning algorithm, described in Chapter 4;
- *gradient_loc.py*: gradient-based optimization methods for localization of users;
- *filtering.py* and *tinyekf.py*: algorithms for signal filtering and processing.

- **Helper** module consists of various helper functions for the simulation.

- **Data Collection** module collects the following main statistics:
  - history of locations for UEs and ABSs;
  - history of RSS, SINR and downlink capacity for UEs;
  - history of aggregated capacity and allocated resource blocks for ABSs;
  - history of execution time and internal state of algorithms for ABSs;
  - history of aggregated system capacity for all ABSs in playground.

  After each simulation run, collected data is serialized and stored in the *results* folder. The simulator also provides multiple postprocessing scripts for simplified data analysis in the *postprocessing* folder.

A detailed description of the simulator and its features can be found in documentation, located in the *doc* folder.

A.2.4 Simulation Method and Constant Simulation Parameters

We use the Monte Carlo method for simulation, relying on repeated random sampling to obtain numerical results [50]. For each of the simulation scenarios, at least 64 runs are processed and the results are averaged. The simulation parameters are summarized in Table A.1.

Table A.1. – Main simulation parameters.

Parameter	**Value/Name**
Carrier frequency	1800 MHz
System bandwidth	1.25 MHz, 6 RBs
ABS transmit power	20 dBm
ABS and UE antenna gains	1
Frequency reuse between ABSs	$FR = 1$
ABS scheduler	r-best-CQI ($r = 6$)
Propagation model	Air-to-Ground [9], ($a = 46$, $b = -2$), ($\gamma = 2.5$, $\sigma = 7.0$)
Traffic model	Full buffer
Area size	2000x2000 m^2
RPGMM user mobility	$v_{min} = 0.5$ m/s, $v_{max} = 2$ m/s, $r_{max} = \alpha$
ABS maximum speed	4 m/s
ABS acceleration	1 m/s^2
Monte-Carlo simulation runs	64
Max. simulation time	1600 sec
Simulation time step	0.1 sec
RSS arrival rate	0.5 sec
Node distribution	Random uniform

A.3 Design of the ABS prototype

In this section we provide a more detailed description of the developed SkySAIL platform.

A.3.1 Processing Unit

We have used a customized Linux-based quadcopter as an MAV. It consists of the ODROID-U3 Single-Board Computer (SBC), an ARM-based low cost and high-performance development platform [89]. This SBC was chosen due to its small size and high processing power. ODROID-U3 is smaller than a credit card, but it has a powerful 1.7 GHz Quad-Core processor and 2 GB RAM. It contains several interfaces, such as General-Purpose Input/Output (GPIO) interface, Universal Asynchronous Receiver-Transmitter (UART), Inter-Integrated Circuit (I2C), USB 2.0, 10/100M Ethernet, satisfying the needs to connect additional hardware.

A.3.2 Operating and Flight Management Systems

The SBC is controlled by the custom Gentoo Linux OS, allowing to run nearly any application, including software for SDR platforms. Besides other applications, it is running a Flight Management System (FMS) - PenguPilot. The FMS includes an autopilot that allows to take off, land and fly autonomously along a predefined or dynamically adapted trajectory. More details about PenguPilot FMS can be found in [90, 91].

A.3.3 Additional components

For navigation purposes, the SBC is connected to multiple sensors, including accelerometer, gyroscope, magnetometer and a GNSS receiver.

Specifically, we have chosen the GNSS receiver from u-blox (type: LEA-6H), which is connected via USB port. The maximum navigation update rate of the GNSS receiver is 5 Hz and the highest horizontal position accuracy: with Satellite-Based Augmentation System (SBAS) is 2.0 m; without aiding - 2.5 m [92].

The MAV is equipped with a 4-cell Vislero lithium polymer battery with a nominal voltage of 14.8 V and capacity of 3300 mAh. This battery allows the MAV to fly up to 24 minutes. With average speed of 2 m/s, the MAV can cover a distance up to 1800 m.

The software running on the SBC, allows a user to share data between the FMS and other applications. This gives the advantage of using real-time data for user applications. For instance, this feature enables the SkySAIL to use a precise, time-accurate location and radio information as an input for the optimization problem.

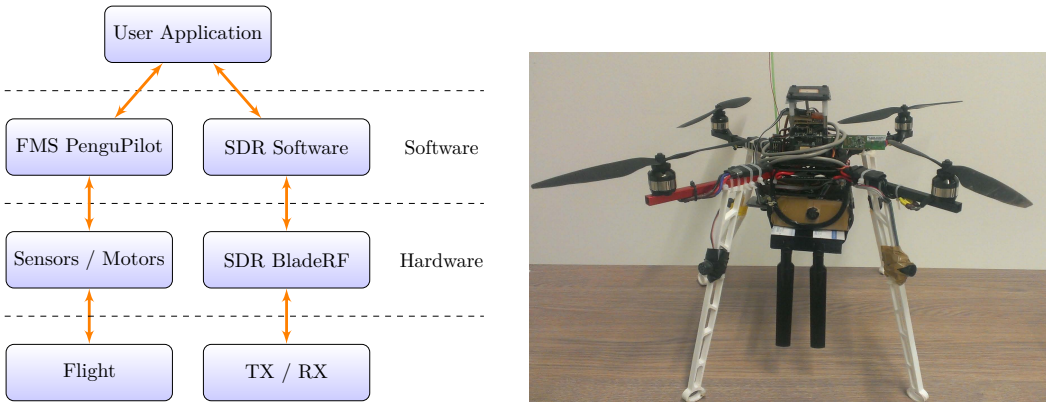
A.3.4 Software-Defined Radio

As it was mentioned earlier, not all SDRs can be installed on MAVs because of their size and weight. Moreover, the SDR should support full-duplex mode and a wide range of frequencies to satisfy the requirements of many applications.

The type of SDR used for this project is bladeRF x40 made by Nuand, California [79]. BladeRF has powerful Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) that are performed in Altera Cyclone IV Field-Programmable Gate Array (FPGA) along with the 200 MHz ARM9 processor. The FPGA reduces the computational complexity and can offload computationally intensive functions from the CPU e.g digital signal processing.

A fully integrated wideband Radio Frequency (RF) Transceiver (LimeMicro LMS6002D) is capable to operate at RF in the range from 300 MHz to 3.8 GHz. A variety of open-source software allows applying the bladeRF as a custom RF modem, a GSM and LTE picocell, a GPS transmitter or a IEEE 802.11/Bluetooth interface without the need for any extra hardware.

The general SkySAIL architecture is shown in Fig. A.6(a). Depending on the target mission, different software can be uploaded and executed on the SkySAIL. Moreover, the software can be changed on the fly, depending on user preferences. The SkySAIL prototype is shown in Fig. A.6(b).



(a) General architecture

(b) Experimental prototype

Figure A.6. – SkySAIL platform.

List of Acronyms

Notation	Description
2D	Two-dimensional space
3D	Three-dimensional space
A2G	Air-to-Ground
ABS	Aerial Base Station
ABSSim	Aerial Base Station Simulator
ADC	Analog-to-Digital Converter
AIDA	Autonomous Infrastructure Deployment Algorithm
ARM	Advanced RISC Machine
AWGN	Additive White Gaussian Noise
BS	Base Station
CPU	Central Processing Unit
CQI	Channel Quality Indicator
DAC	Digital-to-Analog Converter
E-UTRA	Evolved Universal Mobile Telecommunications System
FMDP	Finite Markov Decision Process
FMS	Flight Management System
FPGA	Field-Programmable Gate Array
FSPL	Free Space Path Loss
G2G	Ground-to-Ground
GNSS	Global Navigation Satellite System
GPIO	General-Purpose Input/Output
GPRS	General Packet Radio Services
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphical User Interface

Notation	Description
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMSI	International Mobile Subscriber Identity
ITU-T	International Telecommunication Union
KS	Kolmogorov–Smirnov
LiPo	Lithium-Ion Polymer
LoS	Line-of-Sight
LTE	Long-Term Evolution
MAS	Multi-Agent System
MAV	Micro-Aerial Vehicle
MCS	Modulation and Coding Schemes
NLoS	Non-Line-of-Sight
NP	Non-deterministic Polynomial time
NP-complete	Non-deterministic Polynomial-time complete
NP-hard	Non-deterministic Polynomial-time hard
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OLoS	Obstructed-Line-of-Sight
OS	Operating System
PBX	Private Branch Exchange
PC	Projected Clustering
PC-Quality	Projected Clustering Quality
PF	Proportional Fair
PSO	Particle Swarm Optimization
QoS	Quality of Service
RAM	Random Access Memory
RB	Resource Block
RF	Radio Frequency
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNG	Random Number Generator
RPGMM	Reference Point Group Mobility Model
RR	Round Robin
RRC	Radio Resource Control
RS	Random Search
RSS	Received Signal Strength

Notation	Description
RXLEV_DL	Received Level Downlink
SACCH	Slow Associated Control Channel
SBAS	Satellite-Based Augmentation System
SBC	Single-Board Computer
SDCCH	Standalone Dedicated Control Channel
SDR	Software-Defined Radio
SINR	Signal-to-Interference-and-Noise Ratio
SIP	Session Initiation Protocol
SkySAIL	Software-defined radio enabled micro AerIal vehicLe
SMPP	Short Message Service Point-to-Point
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SO	Self-Organized
TCH	Traffic Channel
TDMA	Time-Division Multiple Access
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
VF	Virtual Fields
VoIP	Voice-Over-IP
VTOL	Vertical Take-Off and Landing

List of Symbols

Notation	Description
B	The size of the target area, where the user distribution should be generated.
C_{total}	The aggregated system capacity in bits per second
C	The maximum communication capacity according to Shannon equation in bits per second
D	The separation threshold in dBm
F	The force vector for artificial field method
G_r	The gain of receiver antenna
G_t	The gain of transmitter antenna
H^g_{best}	The position of the best-performing particle of a topological neighborhood H^g_{best} in the PSO algorithm
H^l_{best}	The best previously visited position H^l_{best} of the particle in the PSO algorithm
H^l	The position of particle H^l in the search space in the PSO algorithm
I	The received power of each interfering signal in W
K_B	The Boltzmann constant in J/K
K	The dataset length
N	The power of thermal noise in W
PL_{LoS}	The individual path loss for LoS component
PL_{NLoS}	The individual path loss for NLoS component
PL_{OLoS}	The individual path loss for OLoS component
PL_{d_0}	The reference path loss at the reference distance d_0
PL	The total pathloss in obstructed environment
P_t	The transmitted power of signal W
Q	The table, which defines the expected reward value of each action in each state in Q-Learning
R_e	The Earth radius, equal to 6371 km (for the WGS84 ellipsoid)
R	The resource allocation scheme for the j-th ABS
$SINR$	The power of noise and interfering signals in W
S	The received power of signal in W
T_k	The temperature in K

Notation	Description
T_x	The x coordinate of the aircraft
T_y	The y coordinate of the aircraft
T_{x0}	The previous x coordinate of the aircraft
T_{y0}	The previous y coordinate of the aircraft
U	The repulsive potential for artificial field method
V^l	The velocity of the particle in swarm in the PSO algorithm
W	The total channel bandwidth in Hz
Z	The cluster center (centroid), which are corresponding to the mean of the user locations in a cluster (Chapter 2)
Z	Set of k centers (centroids) the K-Means algorithm (Chapter 3)
ΔT_x	The x displacement of the aircraft in the body frame
ΔT_y	The y displacement of the aircraft in the body frame
$\Delta\omega$	The turning angle of the aircraft
Δt	The elapsed time
Δ	The maximum deviation Δ of user locations from the centroid Z
Γ	The rotation matrix in inertial frame for the rigid body transformation
Ω	The learning rate in Q-Learning
Π	Transition probabilities between $\mathcal{Z}$ and $\mathcal{A}$ in FMDP
Υ	Immediate rewards for actions $\mathcal{A}$ in FMDP
α_0	The ratio of land area covered by buildings to the total land area according to ITU recommendations
α	Normalized sparsity, which defines the normalized deviation of the user positions from the centroid Z. The $\alpha = 1$ corresponds to a uniform user distribution, while $\alpha = 0.1$ results in q^2 dense clusters.
β_0	The mean number of buildings per unit area (buildings/km^2) according to ITU recommendations
χ	The latitude coordinate
δ	The discount factor in Q-Learning
ϵ	The probability of taking random action by the Q-Learning agent at each iteration
γ_0	A variable determining the building height distribution according to ITU recommendations
γ	The path loss exponent
κ	The stochastic component in the PSO algorithm, which is uniformly chosen from the interval $[0, 1]$
$\mathcal{A}$	Set of possible actions in FMDP
$\mathcal{N}$	The normal distribution with mean μ
$\mathcal{Z}$	Set of possible states in FMDP in Chapter 4

Notation	Description
μ	The scaling coefficient for artificial field method
ω	The bearing angle of the aircraft
ϕ	The inertia weight, which is used to control speed of convergence of the PSO algorithm
π	The policy of selecting actions, which maximizes the expected value of the total reward over all successive steps in Q-Learning
ψ	The coefficiens that determine the acceleration of the particle towards its own and/or the swarm goal in the PSO algorithm
ρ	The longitude coordinate
θ_0	The minimum elevation angle for the segmented A2G path loss model, which is equal to 15°
θ	The elevation angle between a transmitter and a receiver is defined as an acute angle in the range of 0° and 90°
υ	The reward in Q-Learning after the action is done
φ	The decay factor for the ϵ-greedy strategy in Q-Learning
$\vec{T}$	The matrix, which consists rotation, followed by translation in inertial frame for the rigid body transformation
$\vec{T}$	The translation vector in inertial frame for the rigid body transformation
ξ	The propagation group for the segmented A2G path loss model
ζ_{th}	The SINR threshold for successful reception of the signal in W
a_U	The acceleration of the aircraft
a_{t+1}	The possible next action in Q-Learning algorithm
a_t	The selected action in Q-Learning algorithm
a	The city-specific Sigmoid curve parameter for A2G model, which defines at which elevation angle the transition from LoS to NLoS conditions occurs
b	The city-specific Sigmoid curve parameter for A2G model, which defines how fast the transition between LoS to NLoS conditions occurs
c_ξ	The frequency-dependent parameters for the segmented excessive A2G path loss model
d_0	The reference distance for the particular antenna
d_s	The spherical distance between the two points
d_ξ	The environment-dependent parameters for the segmented excessive A2G path loss model
d	The distance between transmitter and receiver in m

Notation	Description
k	The number of clusters for the K-Means problem
l	The distance among cluster centroids Z in one dimension
m	The number of ABSs
n	The number of UEs
$p_{LoS}(\theta)$	The LoS probability for A2G propagation model
$p_{NLoS}(\theta)$	The NLoS probability for A2G propagation model
q	The number of clusters in one dimension, where q^2 gives a number of clusters in 2D space
r	The number of available RBs for each BS
s	The set of ABS locations
t	The elapsed time
u	The set of UE locations
v_0	The previous velocity of the aircraft
v	The velocity of the aircraft
z_{t+1}	The new state in Q-Learning algorithm in Chapter 4
z_t	The current state in Q-Learning algorithm in Chapter 4
z	Element of set Z or center (centroid) for the K-Means algorithm (Chapter 3)

List of Figures

List of Tables

Bibliography

[1] Erdelj, Milan ; Natalizio, Enrico ; Chowdhury, Kaushik R. ; Akyildiz, Ian F.: Help from the sky: Leveraging UAVs for disaster management. In: *IEEE Pervasive Computing* 16 (2017), Nr. 1, S. 24–32

[2] Gomez, Karina ; Hourani, Akram ; Goratti, Leonardo ; Riggio, Roberto ; Kandeepan, Sithamparanathan ; Bucaille, Isabelle: Capacity evaluation of aerial LTE base-stations for public safety communications. In: *Networks and Communications (EuCNC), 2015 European Conference on* IEEE, 2015, S. 133–138

[3] Andryeyev, O. ; Rubina, A. ; Golokolenko, O. ; Artemenko, O. ; Mitschele-Thiel, A.: SkySAIL: A Flexible Software-Defined Radio Enabled Micro Aerial Vehicle. In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, 2016, S. 1–6

[4] Andryeyev, Oleksandr: Spectrum Usage Improvement in Cognitive Radio Networks Using Directional Antenna and UAV Capabilities. In: *SPITSE 2014*. Ilmenau, Germany : Technische Universität Ilmenau, 07 2014, S. 20–24

[5] Andryeyev, O. ; Artemenko, O. ; Mitschele-Thiel, A.: Improving the system capacity using directional antennas with a fixed beam on small Unmanned Aerial Vehicles. In: *2015 European Conference on Networks and Communications (EuCNC)*, 2015, S. 139–143

[6] Andryeyev, Oleksandr ; Onus, Umut ; Casas Melo, Victor ; Mitschele-Thiel, Andreas: Experimental Validation of Air-to-Ground Propagation Models for Low-Altitude Platforms. In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), WiDroIt 2019*. Santorini Island, Greece, 05 2019

[7] Andryeyev, Oleksandr ; Mitschele-Thiel, Andreas: Efficiency vs. Accuracy of Aerial Base Station Placement. In: *International Conference on Networked Systems 2019 (NetSys 2019)*. Garching b. München, Germany : IEEE, 03 2019, S. 1–6

[8] Andryeyev, Oleksandr ; Mitschele-Thiel, Andreas: Increasing the Cellular Network Capacity Using Self-Organized Aerial Base Stations. In: *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* ACM, 2017, S. 37–42

[9] Al-Hourani, A. ; Kandeepan, S. ; Lardner, S.: Optimal LAP Altitude for Maximum Coverage. In: *IEEE Wireless Communications Letters* 3 (2014), Dec, Nr. 6, S. 569–572. http://dx.doi.org/10.1109/LWC.2014.2342736. – DOI 10.1109/LWC.2014.2342736. – ISSN 2162–2337

[10] Azari, Mohammad M. ; Rosas, Fernando ; Chiumento, Alessandro ; Ligata, Amir ; Pollin, Sofie: Uplink performance analysis of a drone cell in a random field of ground interferers. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)* IEEE, 2018, S. 1–6

[11] Lyu, Jiangbin ; Zeng, Yong ; Zhang, Rui: UAV-aided offloading for cellular hotspot. In: *IEEE Transactions on Wireless Communications* 17 (2018), Nr. 6, S. 3988–4001

[12] Mozaffari, Mohammad ; Saad, Walid ; Bennis, Mehdi ; Debbah, Mérouane: Unmanned aerial vehicle with underlaid device-to-device commu-

nications: Performance and tradeoffs. In: *IEEE Transactions on Wireless Communications* 15 (2016), Nr. 6, S. 3949–3963

[13] Mozaffari, Mohammad ; Saad, Walid ; Bennis, Mehdi ; Debbah, Mérouane: Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. In: *IEEE Communications Letters* 20 (2016), Nr. 8, S. 1647–1650

[14] Alzenad, Mohamed ; El-Keyi, Amr ; Lagum, Faraj ; Yanikomeroglu, Halim: 3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. In: *IEEE Wireless Communications Letters* 6 (2017), Nr. 4, S. 434–437

[15] Shakhatreh, Hazim ; Khreishah, Abdallah ; Alsarhan, Ayoub ; Khalil, Issa ; Sawalmeh, Ahmad ; Othman, Noor S.: Efficient 3D placement of a UAV using particle swarm optimization. In: *2017 8th International Conference on Information and Communication Systems (ICICS)* IEEE, 2017, S. 258–263

[16] Bor-Yaliniz, R I. ; El-Keyi, Amr ; Yanikomeroglu, Halim: Efficient 3-D placement of an aerial base station in next generation cellular networks. In: *2016 IEEE international conference on communications (ICC)* IEEE, 2016, S. 1–5

[17] Lyu, Jiangbin ; Zeng, Yong ; Zhang, Rui ; Lim, Teng J.: Placement optimization of UAV-mounted mobile base stations. In: *IEEE Communications Letters* 21 (2016), Nr. 3, S. 604–607

[18] Mozaffari, Mohammad ; Saad, Walid ; Bennis, Mehdi ; Debbah, Mérouane: Mobile Internet of Things: Can UAVs provide an energy-efficient mobile architecture? In: *2016 IEEE global communications conference (GLOBECOM)* IEEE, 2016, S. 1–6

[19] Kalantari, Elham ; Yanikomeroglu, Halim ; Yongacoglu, Abbas: On the number and 3D placement of drone base stations in wireless cellular networks. In: *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th* IEEE, 2016, S. 1–6

[20] Mozaffari, Mohammad ; Kasgari, Ali Taleb Z. ; Saad, Walid ; Bennis, Mehdi ; Debbah, Mérouane: Beyond 5G with UAVs: Foundations of a 3D wireless cellular network. In: *IEEE Transactions on Wireless Communications* 18 (2018), Nr. 1, S. 357–372

[21] Sharma, Vishal ; Bennis, Mehdi ; Kumar, Rajesh: UAV-assisted heterogeneous networks for capacity enhancement. In: *IEEE Communications Letters* 20 (2016), Nr. 6, S. 1207–1210

[22] Ghanavi, Rozhina ; Kalantari, Elham ; Sabbaghian, Maryam ; Yanikomeroglu, Halim ; Yongacoglu, Abbas: Efficient 3D aerial base station placement considering users mobility by reinforcement learning. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)* IEEE, 2018, S. 1–6

[23] Bor-Yaliniz, Irem ; El-Keyi, Amr ; Yanikomeroglu, Halim: Spatial configuration of agile wireless networks with drone-BSs and user-in-the-loop. In: *IEEE Transactions on Wireless Communications* 18 (2018), Nr. 2, S. 753–768

[24] Sharma, Vishal ; Sabatini, Roberto ; Ramasamy, Subramanian: UAVs assisted delay optimization in heterogeneous wireless networks. In: *IEEE Communications Letters* 20 (2016), Nr. 12, S. 2526–2529

[25] Zhong, Shuxin ; Qiu, Yu-Xuan ; Ruby, Rukhsana ; Wang, Lu ; Wu, Kaishun: SIDE: Semi-Distributed Mechanical Equilibrium Based UAV Deployment. In: *2018 IEEE 26th International Conference on Network Protocols (ICNP)* IEEE, 2018, S. 270–279

[26] Galkin, Boris ; Kibilda, Jacek ; DaSilva, Luiz A.: Deployment of UAV-mounted access points according to spatial user locations in two-tier cellular networks. In: *Wireless Days (WD), 2016* IEEE, 2016, S. 1–6

[27] Rohde, Sebastian ; Wietfeld, Christian: Interference aware positioning of aerial relays for cell overload and outage compensation. In: *Vehicular Technology Conference (VTC Fall), 2012 IEEE* IEEE, 2012, S. 1–5

[28] Kalantari, Elham ; Bor-Yaliniz, Irem ; Yongacoglu, Abbas ; Yanikomeroglu, Halim: User association and bandwidth allocation for terrestrial and aerial base stations with backhaul considerations. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* IEEE, 2017, S. 1–6

[29] Al-Hourani, A. ; Kandeepan, S. ; Jamalipour, A.: Modeling air-to-ground path loss for low altitude platforms in urban environments. In: *2014 IEEE Global Communications Conference*, 2014. – ISSN 1930–529X, S. 2898–2904

[30] Bostian, C. W. ; Young, A. R.: The application of cognitive radio to coordinated unmanned aerial vehicle (UAV) missions / Virginia Polytechnic Inst. and State Univ. Blacksburg, Tech. Rep. 2011. – Forschungsbericht

[31] Tarasov, Mikhail: *Self-Organizing Network Optimization via Placement of Additional Nodes*, Ilmenau University of Technology, Doctoral Thesis, 2013

[32] Feng, Qixing ; McGeehan, J. ; Tameh, E. K. ; Nix, A. R.: Path Loss Models for Air-to-Ground Radio Channels in Urban Environments. In: *2006 IEEE 63rd Vehicular Technology Conference* Bd. 6, 2006. – ISSN 1550–2252, S. 2901–2905

[33] Goldsmith, Andrea: *Wireless communications*. Cambridge university press, 2005

[34] 3GPP: *Evolved Universal Terrestrial Radio Access (EUTRA); Medium Access Control (MAC) Protocol Specification, v12.7.0, 2015, TS 36.321.* `http://www.3gpp.org`, accessed on 22 October 2020

[35] Salo, J ; Nur-Alam, M ; Chang, K: Practical introduction to LTE radio planning. In: *A white paper on basics of radio planning for 3GPP LTE in interference limited and coverage limited scenarios, European Communications Engineering (ECE) Ltd, Espoo, Finland* (2010)

[36] Feng, Q. ; Tameh, E. K. ; Nix, A. R. ; McGeehan, J.: WLCp2-06: Modelling the Likelihood of Line-of-Sight for Air-to-Ground Radio Propagation in Urban Environments. In: *IEEE Globecom 2006*, 2006. – ISSN 1930–529X, S. 1–5

[37] Recommendations, ITUR: Propagation data and prediction methods required for the design of terrestrial broadband radio access systems operating in a frequency range from 3 to 60 GHz. In: *ITU-R* (2013)

[38] Motorola: *Long term evolution (lte): Overview of lte air-interface technical white paper.* `https://pdfs.semanticscholar.org/c064/68fe2a2d69e3a58050967876743d272ba929.pdf`, accessed on 22 October 2020

[39] AlQahtani, Salman A. ; AlHassany, Mohammed: Performance modeling and evaluation of novel scheduling algorithm for LTE networks. In: *Network computing and applications (NCA), 2013 12th IEEE international symposium on* IEEE, 2013, S. 101–105

[40] Singh, Davinder ; Singh, Preeti: Radio Resource scheduling in 3GPP LTE: a review. In: *International Journal of Engineering Trends and Technology (IJETT)* 4 (2013), Nr. 6, S. 2405–2411

[41] Müller, Saskia: *Implementation of Resource Scheduling Algorithms for LTE Radio Systems in Python-based Simulator Environment*, Ilmenau University of Technology, Bachelor Thesis, 2018

[42] Schwarz, Stefan ; Mehlführer, Christian ; Rupp, Markus: Low complexity approximate maximum throughput scheduling for LTE. In: *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on* IEEE, 2010, S. 1563–1569

[43] Nousiainen, Sami ; Kordybach, Krzysztof ; Kemppi, Paul: User distribution and mobility model framework for cellular network simulations. In: *VTT Information Technology* (2002), S. 518–522

[44] Young, Ian T.: Proof without prejudice: use of the Kolmogorov-Smirnov test for the analysis of histograms from flow systems and other sources. In: *Journal of Histochemistry & Cytochemistry* 25 (1977), Nr. 7, S. 935–941

[45] Justel, Ana ; Peña, Daniel ; Zamar, Rubén: A multivariate Kolmogorov-Smirnov test of goodness of fit. In: *Statistics & Probability Letters* 35 (1997), Nr. 3, S. 251–259

[46] SciPy: *The Kolmogorov-Smirnov statistic on 2 samples.* `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ks_2samp.html`, accessed on 22 October 2020

[47] Hong, Xiaoyan ; Gerla, Mario ; Pei, Guangyu ; Chiang, Ching-Chuan: A group mobility model for ad hoc wireless networks. In: *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* ACM, 1999, S. 53–60

[48] Bai, Fan ; Helmy, Ahmed: A survey of mobility models. In: *Wireless Adhoc Networks. University of Southern California, USA* 206 (2004), S. 147

[49] Fitzpatrick, Richard: Classical Mechanics: An introductory course. In: *Lulu Enterprises, Inc* (2006)

[50] Kroese, Dirk P. ; Brereton, Tim ; Taimre, Thomas ; Botev, Zdravko I.: Why the Monte Carlo method is so important today. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6 (2014), Nr. 6, S. 386–392

[51] Fowler, Robert J. ; Paterson, Michael S. ; Tanimoto, Steven L.: Optimal packing and covering in the plane are NP-complete. In: *Information processing letters* 12 (1981), Nr. 3, S. 133–137

[52] Erickson, Jeff: *Algorithms and Models of Computation, lecture notes at University of Illinois, Urbana-Champaign, January 2015 revision.* `http://jeffe.cs.illinois.edu/teaching/algorithms/`, accessed on 22 October 2020

[53] Korte, Bernhard ; Vygen, Jens ; Korte, B ; Vygen, J: *Combinatorial optimization.* Bd. 2. Springer, 2012

[54] Nocedal, Jorge ; Wright, Stephen J.: *Numerical optimization 2nd.* 2006

[55] Wright, Margaret H.: Optimization methods for base station placement in wireless applications. In: *VTC'98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)* Bd. 1 IEEE, 1998, S. 387–391

[56] Niculescu, Dragoş ; Nath, Badri: VOR base stations for indoor 802.11 positioning. In: *Proceedings of the 10th annual international conference on Mobile computing and networking* ACM, 2004, S. 58–69

[57] Cortez, Paulo: *Modern optimization with R.* Springer, 2014

[58] Brooks, Samuel H.: A discussion of random methods for seeking maxima. In: *Operations research* 6 (1958), Nr. 2, S. 244–251

[59] *Kapitel* Optimization Methods. In: Rothlauf, Franz: *Optimization Methods.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. – ISBN 978–3–540–72962–4, 45–102

[60] Nelder, John A. ; Mead, Roger: A simplex method for function minimization. In: *The computer journal* 7 (1965), Nr. 4, S. 308–313

[61] Audet, Charles ; Hare, Warren: *Derivative-free and blackbox optimization*. Springer, 2017

[62] Arthur, David ; Vassilvitskii, Sergei: k-means++: The advantages of careful seeding. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* Society for Industrial and Applied Mathematics, 2007, S. 1027–1035

[63] Mahajan, Meena ; Nimbhorkar, Prajakta ; Varadarajan, Kasturi: The planar k-means problem is NP-hard. In: *International Workshop on Algorithms and Computation* Springer, 2009, S. 274–285

[64] Cormen, Thomas H. ; Leiserson, Charles E. ; Rivest, Ronald L. ; Stein, Clifford: *Introduction to algorithms*. MIT press, 2009

[65] Stone, Peter ; Veloso, Manuela: Multiagent systems: A survey from a machine learning perspective. In: *Autonomous Robots* 8 (2000), Nr. 3, S. 345–383

[66] Wooldridge, Michael: *An introduction to multiagent systems*. John Wiley & Sons, 2009

[67] Parker, Lynne E.: Cooperative robotics for multi-target observation. In: *Intelligent Automation & Soft Computing* 5 (1999), Nr. 1, S. 5–19

[68] Parker, Lynne E.: Distributed algorithms for multi-robot observation of multiple moving targets. In: *Autonomous robots* 12 (2002), Nr. 3, S. 231–255

[69] Kerr, Wesley ; Spears, Diana ; Spears, William ; Thayer, David: Two formal gas models for multi-agent sweeping and obstacle avoidance. In: *International Workshop on Formal Approaches to Agent-Based Systems* Springer, 2004, S. 111–130

[70] Winfield, Alan F.: Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In: *Distributed Autonomous Robotic Systems 4*. Springer, 2000, S. 273–282

[71] Pac, Muhammed R. ; Erkmen, Aydan M. ; Erkmen, Ismet: Scalable self-deployment of mobile sensor networks: A fluid dynamics approach. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* IEEE, 2006, S. 1446–1451

[72] Cortes, Jorge ; Martinez, Sonia ; Karatas, Timur ; Bullo, Francesco: Coverage control for mobile sensing networks. In: *IEEE Transactions on robotics and Automation* 20 (2004), Nr. 2, S. 243–255

[73] Tan, Jindong ; Xi, Ning ; Sheng, Weihua ; Xiao, Jizhong: Modeling multiple robot systems for area coverage and cooperation. In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* Bd. 3 IEEE, 2004, S. 2568–2573

[74] Bhattacharya, Subhrajit ; Likhachev, Maxim ; Kumar, Vijay: Multi-agent path planning with multiple tasks and distance constraints. In: *2010 IEEE International Conference on Robotics and Automation* IEEE, 2010, S. 953–959

[75] Long, Pinxin ; Fanl, Tingxiang ; Liao, Xinyi ; Liu, Wenxi ; Zhang, Hao ; Pan, Jia: Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2018, S. 6252–6259

[76] Camp, Tracy ; Boleng, Jeff ; Davies, Vanessa: A survey of mobility models for ad hoc network research. In: *Wireless communications and mobile computing* 2 (2002), Nr. 5, S. 483–502

[77] Watkins, Christopher J. ; Dayan, Peter: Q-learning. In: *Machine learning* 8 (1992), Nr. 3-4, S. 279–292

[78] Weiss, Gerhard: *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 2000

[79] Nuand: *Nuand BladeRF*. `https://www.nuand.com/blog/product/bladerf-x40`, accessed on 22 October 2020

[80] Networks, Range: *OpenBTS Application Suite. User Manual*. `http://openbts.org/documentation/`, accessed on 22 October 2020

[81] Spencer, Mark: *Asterisk Private Branch Exchange: Open Source Communications Software*. `https://www.asterisk.org/`, accessed on 22 October 2020

[82] Institute, European Telecommunications S.: *Digital cellular telecommunications system (Phase 2+); Radio subsystem link control (GSM 05.08)*. 1. `http://www.etsi.org/deliver/etsi_gts/05/0508/05.01.00_60/gsmts_0508v050100p.pdf`, 1996

[83] contributors, OpenStreetMap: *The Kolmogorov-Smirnov statistic on 2 samples*. `https://www.openstreetmap.org/copyright`, accessed on 22 October 2020

[84] Witten, Ian H. ; Frank, Eibe ; Hall, Mark A.: *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2011. – ISBN 0123748569, 9780123748560

[85] Mitchell, Thomas M.: *Machine Learning*. 1. New York, NY, USA : McGraw-Hill, Inc., 1997. – ISBN 0070428077, 9780070428072

[86] Reitermanova, Z: Data splitting. In: *WDS* Bd. 10, 2010, S. 31–36

[87] Onus, Umut: *Air to Ground Propagation Model Parameter Estimation*, Ilmenau University of Technology, Master Thesis, 2019

[88] SimPy, Team: *SimPy, Discrete event simulation for Python*. `https://simpy.readthedocs.io/en/latest/`, accessed on 22 October 2020

[89] *ODROID-U3*. http://www.hardkernel.com/main/products/prdt_info.php?g_code=g138745696275, accessed on 22 October 2020

[90] Jahn, B. ; Barth, A. ; Wulff, K. ; Simon, T. ; Römisch, J.: Rate control and flight stabilization for a quadrotor system. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, S. 642–649

[91] Technology, Ilmenau U.: *Flight Management System (PenguPilot)*. https://github.com/PenguPilot/PenguPilot, accessed on 22 October 2020

[92] u-blox: *U-blox LEA-6H GPS receiver*. https://www.u-blox.com/sites/default/files/products/documents/LEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2FLEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf, accessed on 22 October 2020

[93] Artemenko, O. ; Dominic, O. J. ; Andryeyev, O. ; Mitschele-Thiel, A.: Energy-Aware Trajectory Planning for the Localization of Mobile Devices Using an Unmanned Aerial Vehicle. In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, 2016, S. 1–9

[94] Rubina, Alina ; Artemenko, Oleksandr ; Andryeyev, Oleksandr ; Mitschele-Thiel, Andreas: A Novel Hybrid Path Planning Algorithm for Localization in Wireless Networks. In: *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. New York, NY, USA : ACM, 2017 (DroNet '17). – ISBN 978–1–4503–4960–4, 13–16

[95] Golokolenko, Oleg: *Software Defined Radio Based Sensing on Small Unmanned Aerial Vehicles*, Ilmenau University of Technology, Master Thesis, 2016

[96] Mwanje Dr.-Ing., Stephen S.: *Coordinating Coupled Self-Organized Network Functions in Cellular Radio Networks*, Ilmenau University of Technology, Diss., May 2015. https://www.db-thueringen.de/receive/dbt_mods_00026057

[97] Simon Dr.-Ing., Tobias: *A Self-Organized Unmanned Aerial Message Ferrying System*, Ilmenau University of Technology, Diss., Mar 2015. `https://www.db-thueringen.de/receive/dbt_mods_00025692`

[98] Mozaffari, Mohammad ; Saad, Walid ; Bennis, Mehdi ; Debbah, Merouane: Drone small cells in the clouds: Design, deployment and performance analysis. In: *Global Communications Conference (GLOBECOM), 2015 IEEE* IEEE, 2015, S. 1–6

[99] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Prettenhofer, P. ; Weiss, R. ; Dubourg, V. ; Vanderplas, J. ; Passos, A. ; Cournapeau, D. ; Brucher, M. ; Perrot, M. ; Duchesnay, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830

[100] Jones, Eric ; Oliphant, Travis ; Peterson, Pearu u. a.: *SciPy: Open source scientific tools for Python*. `http://www.scipy.org/`. Version: 2001–. – [Online; accessed 10.11.2018]

[101] Oliphant, Travis E.: *SciPy Tutorial*. `https://www.tau.ac.il/~kineret/amit/scipy_tutorial/`. – [Online; accessed 31.10.2018]

[102] Massaron, Luca ; Boschetti, Alberto: *Regression Analysis with Python*. Packt Publishing Ltd, 2016

[103] Chu, Eunmi ; Yoon, Janghyuk ; Jung, Bang C.: A Novel Link-to-System Mapping Technique Based on Machine Learning for 5G/IoT Wireless Networks. In: *Sensors* 19 (2019), Nr. 5, S. 1196

[104] Rubina, Alina ; Andryeyev, Oleksandr ; Harounabadi, Mehdi ; Al-Khani, Ammar ; Artemenko, Oleksandr ; Mitschele-Thiel, Andreas:

Investigation and Adaptation of Signal Propagation Models for a Mixed Outdoor-Indoor Scenario Using a Flying GSM Base Station. In: ZHOU, Yifeng (Hrsg.) ; KUNZ, Thomas (Hrsg.): *Ad Hoc Networks*. Cham : Springer International Publishing, 2017. – ISBN 978–3–319–51204–4, S. 128–139

[105] ITU-R: Recommendation ITU-R P.676-8: Attenuation by atmospheric gases / International Telecommunication Union: Radiocommunication Sector. 2019 (P.676-8). – Recommendation, P Series

[106] RAPPAPORT, Theodore: *Wireless Communications: Principles and Practice*. 2nd. Upper Saddle River, NJ, USA : Prentice Hall PTR, 2001. – ISBN 0130422320

[107] FRIIS, H. T.: A Note on a Simple Transmission Formula. In: *Proceedings of the IRE* 34 (1946), May, Nr. 5, S. 254–256. http://dx.doi.org/10.1109/JRPROC.1946.234568. – DOI 10.1109/JRPROC.1946.234568. – ISSN 0096–8390

[108] UNION, International T.: *P.1410: Propagation data and prediction methods required for the design of terrestrial broadband radio access systems operating in a frequency range from 3 to 60 GHz*. https://www.itu.int/rec/R-REC-P.1410/en, 2015

[109] 3GPP: Requirements for evolved utra (e-utra) and evolved utran (e-utran). TR 25.913 V7.3.0 / Virginia Polytechnic Inst. and State Univ. Blacksburg, Tech. Rep. 2009 (TR 25.913 V7.3.0). – Forschungsbericht. – 3GPP Standart

[110] MIAO, Guowang ; ZANDER, Jens ; SUNG, Ki W. ; SLIMANE, Slimane B.: *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016

[111] KWAN, Raymond ; LEUNG, Cyril ; ZHANG, Jie: Proportional fair multiuser scheduling in LTE. In: *IEEE Signal Processing Letters* 16 (2009), Nr. 6, S. 461–464

[112] KAWSER, Mohammad T. ; HAMID, Nafiz Imtiaz B. ; HASAN, Md N. ; ALAM, M S. ; RAHMAN, M M.: Downlink snr to cqi mapping for different

multipleantenna techniques in lte. In: *International Journal of Information and Electronics Engineering* 2 (2012), Nr. 5, S. 757

[113] Kawser, Mohammad T. ; Hamid, Nafiz Imtiaz B. ; Hasan, Md N. ; Alam, M S. ; Rahman, M M.: Downlink snr to cqi mapping for different multipleantenna techniques in lte. In: *International Journal of Information and Electronics Engineering* 2 (2012), Nr. 5, S. 757

[114] Shah, Rahul C. ; Roy, Sumit ; Jain, Sushant ; Brunette, Waylon: Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. In: *Ad Hoc Networks* 1 (2003), Nr. 2-3, S. 215–233

[115] Kumar, Santosh ; Sharma, SC ; Suman, Bhupendra: Classification and evaluation of mobility metrics for mobility model movement patterns in mobile ad-hoc networks. In: *International journal on applications of graph theory in wireless ad hoc networks and sensor networks* 3 (2011), Nr. 3, S. 25

[116] Paul, B.: *Kinematics and Dynamics of Planar Machinery*. Prentice-Hall, NJ, 1979

[117] Knuth, Donald E.: *The Art of Computer Programming: Fundamental Algorithms. Redwood City*. 1997

[118] Bostian, Charles W. ; Young, Alexander R.: The application of cognitive radio to coordinated unmanned aerial vehicle (uav) missions / VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY BLACKSBURG. 2011 (AFRL-RI-RS-TR-2011-148). – Forschungsbericht. – Final Technical Report

[119] Huo, J. ; Xu, Z. ; Zhang, Y. ; Shan, X.: A UAV mobile strategy in mobile ad hoc networks. In: *2011 International Conference on Electronics, Communications and Control (ICECC)*, 2011, S. 686–690

[120] Grøtli, Esten I. ; Johansen, Tor A.: Path-and data transmission planning for cooperating UAVs in delay tolerant network. In: *2012 IEEE Globecom Workshops* IEEE, 2012, S. 1568–1573

[121] Han, Zhu ; Swindlehurst, A L. ; Liu, KJ R.: Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles. In: *IEEE Transactions on Vehicular Technology* 58 (2009), Nr. 7, S. 3533–3546